C 8

```
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDDDDDDDDDDD     RRRRRRRRRRRR        IIIIIIIII     VVV         VVV  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRR         RRR        III        VVV         VVV  EEE              RRR         RRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRRRRRRRRRRR           III        VVV         VVV  EEEEEEEEEEEE     RRRRRRRRRRRR
DDD         DDD  RRR   RRR              III        VVV         VVV  EEE              RRR   RRR
DDD         DDD  RRR   RRR              III        VVV         VVV  EEE              RRR   RRR
DDD         DDD  RRR   RRR              III        VVV         VVV  EEE              RRR   RRR
DDD         DDD  RRR      RRR           III           VVV   VVV     EEE              RRR      RRR
DDD         DDD  RRR      RRR           III           VVV   VVV     EEE              RRR      RRR
DDD         DDD  RRR      RRR           III           VVV   VVV     EEE              RRR      RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII           VVV        EEEEEEEEEEEEEEE  RRR         RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII           VVV        EEEEEEEEEEEEEEE  RRR         RRR
DDDDDDDDDDDD     RRR         RRR     IIIIIIIII           VVV        EEEEEEEEEEEEEEE  RRR         RRR
```

B 7

```
**FILE**ID**DBDRIVER



DDDDDDDD    BBBBBBBB    DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
DDDDDDDD    BBBBBBBB    DDDDDDDD    RRRRRRRR      IIIIII    VV      VV  EEEEEEEEEE  RRRRRRRR
DD      DD  BB      BB  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
PD      DD  BB      BB  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD  BB      BB  DD      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD  BB      BB  DP      DD  RR      RR      II      VV      VV  EE          RR      RR
DD      DD  BBBBBBBB    DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRRRR
DD      DD  BBBBBBBB    DD      DD  RRRRRRRR        II      VV      VV  EEEEEEEE    RRRRRPRR
DD      DD  BB      BB  DD      DD  RR  RR          II      VV      VV  EE          RR  RR
DD      DD  BB      BB  DD      DD  RR  RR          II      VV      VV  EE          RR  RR
DD      DD  BB      BB  DD      DD  RR    RR        II        VV  VV    EE          RR    RR        ....
DD      DD  BB      BB  DD      DD  RR    RR        II        VV  VV    EE          RR    RR        ....
DDDDDDDD    BBBBBBBB    DDDDDDDD    RR      RR    IIIIII        VV      EEEEEEEEEE  RR      RR      ....
DDDDDDDD    BBBBBBBB    DDDDDDDD    RR      RR    IIIIII        VV      EEEEEEEEEE  RR      RR      ....


LL          IIIIII      SSSSSSSS
LL          IIIIII      SSSSSSSS
LL            II      SS
LL            II      SS
LL            II      SS
LL            II      SS
LL            II        SSSSSS
LL            II        SSSSSS
LL            II              SS
LL            II              SS
LL            II              SS
LL            II              SS
LLLLLLLLLL  IIIIII    SSSSSSSS
LLLLLLLLLL  IIIIII    SSSSSSSS
```

DBD
V04

Table of contents

```
0000          .TITLE  DBDRIVER - RP04/05/06 DISK DRIVER
0000          .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000
0000    ; D. N. CUTLER 30-JAN-77
0000
0000    ; MODIFIED BY:
0000
0000    ;       V03-012 RAS0300         Ron Schaefer            27-Apr-1984
0000    ;               Add DEV$M_NNM characteristic to DECHAR2 so that these
0000    ;               devices will have the "node$" prefix.
0000    ;
0000    ;       V03-011 PRD0074         Paul R. DeStefano       28-Feb-1984
0000    ;               Modified ERROR routine so that software volume valid
0000    ;               isn't set if a pack acknowledge function is executed
0000    ;               and medium online isn't set.
0000    ;
0000    ;       V03-010 PRD0031         Paul R. DeStefano       09-Sep-1983
0000    ;               Added EXE$LCLDSKVALID to function decision table.
0000    ;
0000    ;       V03-009 ROW0211         Ralph O. Weber          16-AUG-1983
0000    ;               Change device-dependent UCB definition base from UCB$W_BCR to
0000    ;               UCB$K_LCL_DISK_LENGTH.  Also change UCB$L_DB_BCR to overlay
0000    ;               UCB$L_BCR, a field newly created to meet the needs of this
0000    ;               driver.
0000    ;
0000    ;       V03-008 PRD0022         Paul R. DeStefano       05-May-1983
0000    ;               Modified ERROR routine to attempt to clear a drive
0000    ;               unsafe condidtion.
0000    ;
0000    ;       V03-007 PRD53302        Paul R. DeStefano       04-May-1983
0000    ;       ECO 02  Modified RETRYERR routine to issue a Drive Clear before
0000    ;               retrying a function.  Modified FUNCXT routine to issue
0000    ;               a Drive Clear function before releasing the drive.
```

```
0000    ;
0000    ;       V03-006 PRD0017         Paul R. DeStefano       26-Apr-1983
0000    ;               Modified FATALERR routine to return SS$_PARITY only for
0000    ;               errors that possibly indicate bad media.  All other error
0000    ;               conditions which formerly returned SS$_PARITY now return
0000    ;               SS$_CNTLERR.
0000    ;
0000    ;       V03-005 PRD0012         Paul R. DeStefano       14-Apr-1983
0000    ;               Modified ECC correction logic so that ECC is only applied
0000    ;               when there is single bit ECC correctable error, or if there
0000    ;               is a multiple bit ECC correctable error and the error cannot
0000    ;               be corrected using retries.
0000    ;
0000    ;       V03-04  ROW47161        Ralph O. Weber          17-SEP-1982
0000    ;       ECO 01  Enhance ECC recovery logic to prevent bytes transfered counts
0000    ;               which are not exact multiples of 512 from causing transfer
0000    ;               parameters from being incorrectly updated.  Because a non-512-
0000    ;               intergal bytes transfered counts indicates an incomplete
0000    ;               transfer of the last block, this change also prevents ECC
0000    ;               corrections when such bytes transfered counts are encountered.
0000    ;
0000    ;       V03-003 KDM0002         Kathleen D. Morse       28-Jun-1982
0000    ;               Added $DCDEF, $DYNDEF, $PRDEF, and $SSDEF.
0000    ;
0000    ;       V03-002 KTA0100         Kerbey T. Altmann               07-Jun-1982
0000    ;               Add code to set UCB$L_MEDIA_ID field.
0000    ;
0000    ;
0000    ;**
0000    ;
0000    ; RP04/04/06 DISK DRIVER
0000    ;
0000    ; MACRO LIBRARY CALLS
0000    ;
0000
0000            $CRBDEF                         ;DEFINE CRB OFFSETS
0000            $DCDEF                          ;DEFINE DEVICE CLASSES
0000            $DEVDEF                         ;DEFINE DEVICE CHARACTERISTICS BITS
0000            $DDBDEF                         ;DEFINE DDB OFFSETS
0000            $DPTDEF                         ;DEFINE DPT OFFSETS
0000            $DYNDEF                         ;DEFINE DYNAMIC DATA STRUCTURE TYPES
0000            $EMBDEF                         ;DEFINE EMB OFFSETS
0000            $IDBDEF                         ;DEFINE IDB OFFSETS
0000            $IODEF                          ;DEFINE I/O FUNCTION CODES
0000            $IRPDEF                         ;DEFINE IRP OFFSETS
0000            $MBADEF                         ;DEFINE MBA REGISTER OFFSETS
0000            $PRDEF                          ;DEFINE PROCESSOR REGISTERS
0000            $SSDEF                          ;DEFINE SYSTEM STATUS CODES
0000            $UCBDEF                         ;DEFINE UCB OFFSETS
0000            $VECDEF                         ;DEFINE INTERRUPT DISPATCH VECTOR OFFSETS
0000
0000    ;
0000    ; LOCAL MACROS
0000    ;
0000    ; EXECUTE FUNCTION AND BRANCH ON RETRIABLE ERROR CONDITION
0000    ;
0000
```

```
0000          .MACRO  EXFUNC BDST,FCODE
0000                  .IF NB  FCODE
0000                  MOVZBL  #CD'FCODE,R0
0000                  .ENDC
0000                  BSBW    FEX
0000                  .SIGNED_WORD BDST-.-2
0000          .ENDM
0000
0000  ;
0000  ; GENERATE FUNCTION TABLE ENTRY AND CASE TABLE INDEX SYMBOL
0000  ;
0000
0000          .MACRO  GENF FCODE
0000                  CD'FCODE=.-FTAB
0000                  .BYTE   FCODE!RP_CS1_M_GO
0000          .ENDM
0000
0000  ;
0000  ; LOCAL SYMBOLS
0000  ;
0000  ; RP04/05/06 MASSBUS REGISTER OFFSETS
0000  ;
0000
0000          $DEFINI RP
0000
0000  $DEF    RP_CS1          .BLKL   1       ;DRIVE CONTROL REGISTER
0004          _VIELD  RP_CS1,0,<-             ; DRIVE CONTROL REGISTER BIT DEFINITIONS
0004                  <GO,,M>,-               ; GO BIT
0004                  <FCODE,5>-              ; FUNCTION CODE
0004          >                               ;
0004  $DEF    RP_DS           .BLKL   1       ;DRIVE STATUS REGISTER
0008          _VIELD  RP_DS,6,<-              ; DRIVE STATUS REGISTER BIT DEFINITIONS
0008                  <VV,,M>,-               ; VOLUME VALID
0008                  <DRY,,M>,-              ; DRIVE READY
0008                  <DPR,,M>,-              ; DRIVE PRESENT
0008                  <PGM,,M>,-              ; PROGRAMMABLE
0008                  <LST,,M>,-              ; LAST SECTOR TRANSFERED
0008                  <WRL,,M>,-              ; DRIVE WRITE LOCKED
0008                  <MOL,,M>,-              ; MEDIUM ONLINE
0008                  <PIP,,M>,-              ; POSITIONING IN PROGRESS
0008                  <ERR,,M>,-              ; COMPOSITE ERROR
0008                  <ATA,,M>-               ; ATTENTION ACTIVE
0008          >                               ;
0008  $DEF    RP_ER1          .BLKL   1       ;ERROR REGISTER 1
000C          _VIELD  RP_ER1,0,<-             ; ERROR REGISTER 1 BIT DEFINITIONS
000C                  <ILF,,M>,-              ; ILLEGAL FUNCTION
000C                  <ILR,,M>,-              ; ILLEGAL REGISTER
000C                  <RMR,,M>,-              ; REGISTER MODIFY REFUSED
000C                  <PAR,,M>,-              ; PARITY ERROR
000C                  <FER,,M>,-              ; FORMAT ERROR
000C                  <WCF,,M>,-              ; WRITE CLOCK FAIL
000C                  <ECH,,M>,-              ; ECC HARD ERROR
000C                  <HCE,,M>,-              ; HEADER COMPARE ERROR
000C                  <HCRC,,M>,-             ; HEADER CRC ERROR
000C                  <AOE,,M>,-              ; ADDRESS OVERFLOW ERROR
000C                  <IAE,,M>,-              ; ILLEGAL ADDRESS ERROR
000C                  <WLE,,M>,-              ; WRITE LOCK ERROR
```

```
          000C                       <DTE,,M>,-                    ; DRIVE TIMING ERROR
          000C                       <OPI,,M>,-                    ; OPERATION INCOMPLETE
          000C                       <UNS,,M>,-                    ; DRIVE UNSAFE
          000C                       <DCK,,M>-                     ; DATA CHECK ERROR
          000C              >                                      ;
          000C  $DEF        RP_MR               .BLKL   1          ;MAINTENANCE REGISTER
          0010  $DEF        RP_AS               .BLKL   1          ;ATTENTION SUMMARY REGISTER
          0014  $DEF        RP_DA               .BLKL   1          ;DESIRED SECTOR/TRACK ADDRESS REGISTER
          0018              _VIELD   RP_DA,0,<-                    ; DESIRED ADDRESS FIELD DEFINITIONS
          0018                       <SA,5>,-                      ; DESIRED SECTOR ADDRESS
          0018                       <,3>,-                        ; RESERVED BITS
          0018                       <TA,5>-                       ; DESIRED TRACK ADDRESS
          0018              >                                      ;
          0018  $DEF        RP_DT               .BLKL   1          ;DRIVE TYPE REGISTER
          001C              _VIELD   RP_DT,0,<-                    ; DRIVE TYPE REGISTER FIELD DEFINITIONS
          001C                       <DTN,9>,-                     ; DRIVE TYPE NUMBER
          001C                       <,2>,-                        ; RESERVED BITS
          001C                       <DRQ,,M>-                     ; DRIVE REQUEST REQUIRED
          001C              >                                      ;
          001C  $DEF        RP_LA               .BLKL   1          ;LOOKAHEAD REGISTER
          0020  $DEF        RP_ER2              .BLKL   1          ;ERROR REGISTER 2
          0024  $DEF        RP_OF               .BLKL   1          ;OFFSET REGISTER
          0028              _VIELD   RP_OF,0,<-                    ; OFFSET REGISTER BIT DEFINITIONS
          0028                       <OFF,8>,-                     ; OFFSET VALUE
          0028                       <DCK,,M>,-                    ; DATA CHECK IN PROGRESS (SOFTWARE)
          0028                       <,1>,-                        ; RESERVED BIT
          0028                       <HCI,,M>,-                    ; HEADER COMPARE INHIBIT
          0028                       <ECI,,M>,-                    ; ECC INHIBIT
          0028                       <FMT,,M>-                     ; 16-BIT FORMAT
          0028              >                                      ;
          0028  $DEF        RP_DC               .BLKL   1          ;DESIRED CYLINDER ADDRESS
          002C  $DEF        RP_CC               .BLKL   1          ;CURRENT CYLINDER ADDRESS
          0030  $DEF        RP_SN               .BLKL   1          ;DRIVE SERIAL NUMBER
          0034  $DEF        RP_ER3              .BLKL   1          ;ERROR REGISTER 3
          0038              _VIELD   RP_ER3,14,<-                  ; ERROR REGISTER 3 BIT DEFINITIONS
          0038                       <SKI,,M>-                     ; SEEK INCOMPLETE
          0038              >                                      ;
          0038  $DEF        RP_EC1              .BLKL   1          ;ECC POSITION REGISTER
          003C              _VIELD   RP_EC1,0,<<POS,13>>           ; ECC POSITION FIELD
          003C  $DEF        RP_EC2              .BLKL   1          ;ECC PATTERN REGISTER
          0040              _VIELD   RP_EC2,0,<<PAT,11>>           ; ECC PATTERN FIELD
          0040
          0040              $DEFEND RP
          0000
          0000  ;
          0000  ; DEFINE DEVICE DEPENDENT UNIT CONTROL BLOCK OFFSETS
          0000  ;
          0000
          0000              $DEFINI UCB
          0000
000000CC  0000  .=UCB$K_LCL_DISK_LENGTH                            ; Establish device-dependent UCB base
          00CC
000000C0  00CC  UCB$L_DB_BCR = UCB$L_BCR                           ; Local BCR longword overlays the
          00CC                                                     ; space reserved in the UCB.
          00CC                                                     ; N.B. most drivers only need a word.
          00CC
          00CC  $DEF        UCB$W_DB_ER3        .BLKW   1          ; Space to save RP_ER3 after operation.
          00CE
          00CE  $DEF     UCB$L_DB_SR      .BLKL   1       ;SAVE MBA STATUS REGISTER
          00D2
          00D2  $DEF     UCB$B_DB_ERL     .BLKB   1       ; Space for flag used to signal Medium
          00D3                                            ;  offline at start of function.
          00D3           _VIELD  ERL,0,<-
          00D3                   <MEDOFF,,M>, -           ; MEDIUM OFFLINE FLAG
          00D3                   <DUALPORT,,M>, -         ; DUALPORT KIT FLAG
          00D3                   <ECC_DEFER,,M>, -        ; Flag to indicate that ECC correction
          00D3           >                                ;  has been deferred until offset
          00D3                                            ;  retries are exhausted.
          00D3
000000D6  00D3                            .BLKB   3       ; Reserved.
000000D6  00D6  UCB$K_DB_LENGTH=.                         ; Length of UCB for DB devices.
          00D6           $DEFEND
          0000
          0000  ;
          0000  ; HARDWARE FUNCTION CODES
          0000  ;
          0000
00000000  0000  F_NOP=0*2                                 ;NO OPERATION
00000002  0000  F_UNLOAD=1*2                              ;UNLOAD DRIVE
00000004  0000  F_SEEK=2*2                                ;SEEK CYLINDER
00000006  0000  F_RECAL=3*2                               ;RECALIBRATE
00000008  0000  F_DRVCLR=4*2                              ;DRIVE CLEAR
0000000A  0000  F_RELEASE=5*2                             ;RELEASE DRIVE
0000000C  0000  F_OFFSET=6*2                              ;OFFSET HEADS
0000000E  0000  F_RETCENTER=7*2                           ;RETURN TO CENTERLINE
00000010  0000  F_READPRESET=8*2                          ;READ IN PRESET
00000012  0000  F_PACKACK=9*2                             ;PACK ACKNOWLEDGE
00000018  0000  F_SEARCH=12*2                             ;SEARCH FOR SECTOR
00000018  0000  F_SEARCHA=12*2                            ;SEARCH AHEAD FOR SECTOR
00000028  0000  F_WRITECHECK=20*2                         ;WRITE CHECK DATA
0000002A  0000  F_WRITECHECKH=21*2                        ;WRITE CHECK HEADER AND DATA
00000030  0000  F_WRITEDATA=24*2                          ;WRITE DATA
00000032  0000  F_WRITEHEAD=25*2                          ;WRITE HEADER AND DATA
00000038  0000  F_READDATA=28*2                           ;READ DATA
0000003A  0000  F_READHEAD=29*2                           ;READ HEADER AND DATA
          0000
          0000  ;
          0000  ; LOCAL DATA
          0000  ;
          0000  ; DRIVER PROLOGUE TABLE
          0000  ;
          0000
          0000           DPTAB   -                        ;DEFINE DRIVER PROLOGUE TABLE
          0000                   END=DB_END,-             ;END OF DRIVER
          0000                   ADAPTER=MBA,-            ;ADAPTER TYPE
          0000                   FLAGS=DPT$M_SVP,-        ;SYSTEM PAGE TABLE ENTRY REQUIRED
          0000                   UCBSIZE=UCB$K_DB_LENGTH,-;UCB SIZE
          0000                   NAME=DBDRIVER            ;DRIVER NAME
          0038           DPT_STORE INIT                   ;CONTROL BLOCK INIT VALUES
          0038           DPT_STORE DDB,DDB$L_ACPD,L,<^A\F11\> ;DEFAULT ACP NAME
          003F           DPT_STORE DDB,DDB$L_ACPD+3,B,DDB$K_PACK ;ACP CLASS
          0043           DPT_STORE UCB,UCB$B_FIPL,B,8     ;FORK IPL
          0047           DPT_STORE UCB,UCB$L_DEVCHAR,L,- ;DEVICE CHARACTERISTICS
          0047                   <DEV$M_FOD-              ; FILES ORIENTED
```

```
              0047                !DEV$M_DIR-             ; DIRECTORY STRUCTURED
              0047                !DEV$M_AVL-             ; AVAILABLE
              0047                !DEV$M_ELG-             ; ERROR LOGGING ENABLED
              0047                !DEV$M_SHR-             ; SHAREABLE
              0047                !DEV$M_IDV-             ; INPUT DEVICE
              0047                !DEV$M_ODV-             ; OUTPUT DEVICE
              0047                !DEV$M_RND>             ; RANDOM ACCESS
              004E        DPT_STORE UCB,UCB$L_DEVCHAR2,L,-; DEVICE CHARACTERISTICS
              004E                <DEV$M_NNM>             ; PREFIX NAME WITH "node$"
              0055        DPT_STORE UCB,UCB$B_DEVCLASS,B,DC$_DISK ;DEVICE CLASS
              0059        DPT_STORE UCB,UCB$W_DEVBUFSIZ,W,512 ;DEFAULT BUFFER SIZE
              005E        DPT_STORE UCB,UCB$B_DIPL,B,21   ;DEVICE IPL
              0062        DPT_STORE UCB,UCB$B_ERTCNT,B,8  ;ERROR RETRY COUNT
              0066        DPT_STORE UCB,UCB$B_ERTMAX,B,8  ;MAX ERROR RETRY COUNT
              006A        DPT_STORE REINIT                ;CONTROL BLOCK RE-INIT VALUES
              006A        DPT_STORE DDB,DDB$L_DDT,D,DB$DDT ;DDT ADDRESS
              006F        DPT_STORE END                   ;
              0000
              0000  ;
              0000  ; DRIVER DISPATCH TABLE
              0000  ;
              0000
              0000        DDTAB   DB,-                    ;DRIVER DISPATCH TABLE
              0000                DB_STARTIO,-            ;START I/O OPERATION
              0000                DB_UNSOLNT,-            ;UNSOLICITED INTERRUPT
              0000                DB_FUNCTABLE,-          ;FUNCTION DECISION TABLE
              0000                0,-                     ;CANCEL I/O ENTRY POINT
              0000                DB_REGDUMP,-            ;REGISTER DUMP ROUTINE
              0000                <<RP_EC2+4+MBA$L_BCR+4+8>+<<3+5+1>*4>>,- ;DIAGNOSTIC BUFFER SIZE
              0000                <<RP_EC2+4+MBA$L_BCR+4+8>+<1*4>+<EMB$L_DV_REGSAV>>,- ;ERROR BUFFER S
              0000                DB_RPOX_INIT            ;UNIT INITIALIZATION
              0038
              0038  ;
              0038  ; DATA CHECK FUNCTION TRANSLATION TABLE
              0038  ;
              0038
              0038  CHECKTAB:                             ;
         0A'  0038        .BYTE   CDF_WRITECHECK          ;WRITE DATA
         0A'  0039        .BYTE   CDF_WRITECHECK          ;READ DATA
         0F'  003A        .BYTE   CDF_WRITECHECKH         ;WRITE HEADER AND DATA
         0F'  003B        .BYTE   CDF_WRITECHECKH         ;READ HEADER AND DATA
              003C
              003C  ;
              003C  ; RPOX DRIVE TYPE DESCRIPTOR TABLE
              003C  ;
              003C
              003C  DB_DTDESC:                            ;
       0010   003C        .WORD   ^X10                    ;RP04
         03   003E        .BYTE   DT$_RP04                ;
         16   003F        .BYTE   22                      ;22 SECTORS
         13   0040        .BYTE   19                      ;19 TRACKS
       019B   0041        .WORD   411                     ;411 CYLINDERS PER PACK
   00029F16   0043        .LONG   411*19*22               ;MAXIMUM BLOCKS PER PACK
   20A50004   0047        .LONG   ^X20A50004              ;MEDIA ID "DB RP04"
   0000000F   004B  DB_DTDESCLEN=.-DB_DTDESC              ;LENGTH OF DRIVE TYPE DESCRIPTOR
       0011   004B        .WORD   ^X11                    ;RP05
         04   004D        .BYTE   DT$_RP05                ;
```

```
      16  004E          .BYTE   22                      ;22 SECTORS
      13  004F          .BYTE   19                      ;19 TRACKS
    019B  0050          .WORD   411                     ;411 CYLINDERS PER PACK
00029F16  0052          .LONG   411*19*22               ;MAXIMUM BLOCKS PER PACK
20A50004  0056          .LONG   ^X20A50004              ;MEDIA ID "DB RP04"
    0012  005A          .WORD   ^X12                    ;RP06
      05  005C          .BYTE   DT$_RP06                ;
      16  005D          .BYTE   22                      ;22 SECTORS
      13  005E          .BYTE   19                      ;19 TRACKS
    032F  005F          .WORD   815                     ;815 CYLINDERS PER PACK
000532BE  0061          .LONG   815*19*22               ;MAXIMUM BLOCKS PER PACK
20A50006  0065          .LONG   ^X20A50006              ;MEDIA ID "DB RP06"
          0069
    0000  0069          .WORD   0                       ;END OF TABLE
0000007A  006B          .BLKB   DB_DTDESCLEN            ;SPARE DRIVE TYPE SLOT
00000089  007A          .BLKB   DB_DTDESCLEN            ;SPARE DRIVE TYPE SLOT
          0089
          0089  ;
          0089  ; HARDWARE I/O FUNCTION CODE TABLE
          0089  ;
          0089
          0089  FTAB:                                   ;
          0089          GENF    F_NOP                   ;NO OPERATION
          008A          GENF    F_UNLOAD                ;UNLOAD VOLUME
          008B          GENF    F_SEEK                  ;SEEK CYLINDER
          008C          GENF    F_RECAL                 ;RECALIBRATE
          008D          GENF    F_DRVCLR                ;DRIVE CLEAR
          008E          GENF    F_NOP                   ;RELEASE PORT (NOP)
          008F          GENF    F_OFFSET                ;OFFSET HEADS
          0090          GENF    F_RETCENTER             ;RETURN HEADS TO CENTERLINE
          0091          GENF    F_PACKACK               ;PACK ACKNOWLEDGE
          0092          GENF    F_SEARCH                ;SEARCH FOR SECTOR
          0093          GENF    F_WRITECHECK            ;WRITE CHECK
          0094          GENF    F_WRITEDATA             ;WRITE DATA
          0095          GENF    F_READDATA              ;READ DATA
          0096          GENF    F_WRITEHEAD             ;WRITE HEADER AND DATA
          0097          GENF    F_READHEAD              ;READ HEADER AND DATA
          0098          GENF    F_WRITECHECKH           ;WRITE CHECK HEADER AND DATA
          0099          GENF    F_READPRESET            ;READ IN PRESET
          009A          GENF    F_SEARCHA               ;SEARCH AHEAD FOR SECTOR
          009B
          009B  ;
          009B  ; OFFSET TABLE FOR RP06 - RP04 VALUES = RP06 VALUES * 2 & ^XFF
          009B  ;
          009B
          009B  OFFTAB:                                 ;
      00  009B          .BYTE   0                       ;RETURN TO CENTERLINE
      08  009C          .BYTE   ^X8                     ;+200 (+400)
      C8  009D          .BYTE   ^XC8                    ;-200 (-400)
      10  009E          .BYTE   ^X10                    ;+400 (+800)
      D0  009F          .BYTE   ^XD0                    ;-400 (-800)
      18  00A0          .BYTE   ^X18                    ;+600 (+1200)
      D8  00A1          .BYTE   ^XD8                    ;-600 (-1200)
      00  00A2          .BYTE   0                       ;RETURN TO CENTERLINE
00000008  00A3  OFFSIZ=.-OFFTAB                         ;SIZE OF OFFSET TABLE
```

```
00A3            .SBTTL  RP04/05/06 FUNCTION DECISION TABLE
00A3    ;+
00A3    ; RP04/05/06 FUNCTION DECISION TABLE
00A3    ;-
00A3
00A3    DB_FUNCTABLE:                                   ;FUNCTION DECISION TABLE
00A3            FUNCTAB ,-                              ;LEGAL FUNCTIONS
00A3                    <NOP,-                          ;NO OPERATION
00A3                     UNLOAD,-                       ;UNLOAD VOLUME
00A3                     SEEK,-                         ;SEEK CYLINDER
00A3                     RECAL,-                        ;RECALIBRATE
00A3                     DRVCLR,-                       ;DRIVE CLEAR
00A3                     RELEASE,-                      ;RELEASE PORT
00A3                     OFFSET,-                       ;OFFSET HEADS
00A3                     RETCENTER,-                    ;RETURN HEADS TO CENTERLINE
00A3                     PACKACK,-                      ;PACK ACKNOWLEDGE
00A3                     SEARCH,-                       ;SEARCH FOR SECTOR
00A3                     READPRESET,-                   ;READ IN PRESET
00A3                     SENSECHAR,-                    ;SENSE CHARACTERISTICS
00A3                     SETCHAR,-                      ;SET CHARACTERISTICS
00A3                     SENSEMODE,-                    ;SENSE MODE
00A3                     SETMODE,-                      ;SET MODE
00A3                     WRITECHECK,-                   ;WRITE CHECK
00A3                     WRITEHEAD,-                    ;WRITE HEADER AND DATA
00A3                     READHEAD,-                     ;READ HEADER AND DATA
00A3                     WRITECHECKH,-                  ;WRITE CHECK HEADER AND DATA
00A3                     READLBLK,-                     ;READ LOGICAL BLOCK
00A3                     WRITELBLK,-                    ;WRITE LOGICAL BLOCK
00A3                     READPBLK,-                     ;READ PHYSICAL BLOCK
00A3                     WRITEPBLK,-                    ;WRITE PHYSICAL BLOCK
00A3                     READVBLK,-                     ;READ VIRTUAL BLOCK
00A3                     WRITEVBLK,-                    ;WRITE VIRTUAL BLOCK
00A3                     AVAILABLE,-                    ;UNIT AVAILABLE
00A3                     ACCESS,-                       ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
00A3                     ACPCONTROL,-                   ;ACP CONTROL FUNCTION
00A3                     CREATE,-                       ;CREATE FILE AND/OR CREATE DIRECTORY ENTRY
00A3                     DEACCESS,-                     ;DEACCESS FILE
00A3                     DELETE,-                       ;DELETE FILE AND/OR DIRECTORY ENTRY
00A3                     MODIFY,-                       ;MODIFY FILE ATTRIBUTES
00A3                     MOUNT>                         ;MOUNT VOLUME
00AB            FUNCTAB ,-                              ;BUFFERED I/O FUNCTIONS
00AB                    <NOP,-                          ;NO OPERATION
00AB                     UNLOAD,-                       ;UNLOAD VOLUME
00AB                     SEEK,-                         ;SEEK CYLINDER
00AB                     RECAL,-                        ;RECALIBRATE
00AB                     DRVCLR,-                       ;DRIVE CLEAR
00AB                     RELEASE,-                      ;RELEASE PORT
00AB                     OFFSET,-                       ;OFFSET HEADS
00AB                     RETCENTER,-                    ;RETURN HEADS TO CENTERLINE
00AB                     PACKACK,-                      ;PACK ACKNOWLEDGE
00AB                     SEARCH,-                       ;SEARCH FOR SECTOR
00AB                     AVAILABLE,-                    ;UNIT AVAILABLE
00AB                     READPRESET,-                   ;READ IN PRESET
00AB                     SENSECHAR,-                    ;SENSE CHARACTERISTICS
00AB                     SETCHAR,-                      ;SET CHARACTERISTICS
00AB                     SENSEMODE,-                    ;SENSE MODE
00AB                     SETMODE,-                      ;SET MODE
```

```
00AB                        ACCESS,-                ;ACCESS FILE AND/OR FIND DIRECTORY ENTRY
00AB                        ACPCONTROL,-            ;ACP CONTROL FUNCTION
00AB                        CREATE,-                ;CREATE FILE AND/OR CREATE DIRECTORY ENTRY
00AB                        DEACCESS,-              ;DEACCESS FILE
00AB                        DELETE,-                ;DELETE FILE AND/OR DIRECTORY ENTRY
00AB                        MODIFY,-                ;MODIFY FILE ATTRIBUTES
00AB                        MOUNT>                  ;MOUNT VOLUME
00B3              FUNCTAB +ACP$READBLK,-            ;READ FUNCTIONS
00B3                      <READHEAD,-               ;READ HEADER
00B3                        READLBLK,-              ;READ LOGICAL BLOCK
00B3                        READPBLK,-              ;READ PHYSICAL BLOCK
00B3                        READVBLK>               ;READ VIRTUAL BLOCK
00BF              FUNCTAB +ACP$WRITEBLK,-           ;WRITE FUNCTIONS
00BF                      <WRITECHECK,-             ;WRITE CHECK
00BF                        WRITECHECKH,-           ;WRITE CHECK HEADER AND DATA
00BF                        WRITEHEAD,-             ;WRITE HEADER
00BF                        WRITELBLK,-             ;WRITE LOGICAL BLOCK
00BF                        WRITEPBLK,-             ;WRITE PHYSICAL BLOCK
00BF                        WRITEVBLK>              ;WRITE VIRTUAL BLOCK
00CB              FUNCTAB +ACP$ACCESS,<ACCESS,CREATE> ;ACCESS AND CREATE FILE OR DIRECTORY
00D7              FUNCTAB +ACP$DEACCESS,<DEACCESS> ;DEACCESS FILE
00E3              FUNCTAB +ACP$MODIFY,-             ;
00E3                      <ACPCONTROL,-             ;ACP CONTROL FUNCTION
00E3                        DELETE,-                ;DELETE FILE OR DIRECTORY ENTRY
00E3                        MODIFY>                 ;MODIFY FILE ATTRIBUTES
00EF              FUNCTAB +ACP$MOUNT,<MOUNT>        ;MOUNT VOLUME
00FB              FUNCTAB +EXE$LCLDSKVALID,-        ;LOCAL DISK VALID FUNCTIONS
00FB                      <UNLOAD,-                 ;UNLOAD VOLUME
00FB                        AVAILABLE,-             ;UNIT AVAILABLE
00FB                        PACKACK>                ;PACK ACKNOWLEDGE
0107              FUNCTAB +EXE$ZEROPARM,-           ;ZERO PARAMETER FUNCTIONS
0107                      <NOP,-                    ;NO OPERATION
0107                        UNLOAD,-                ;UNLOAD VOLUME
0107                        RECAL,-                 ;RECALIBRATE
0107                        DRVCLR,-                ;DRIVE CLEAR
0107                        RELEASE,-               ;RELEASE PORT
0107                        RETCENTER,-             ;RETURN HEADS TO CENTERLINE
0107                        READPRESET,-            ;READ IN PRESET
0107                        AVAILABLE,-             ;UNIT AVAILABLE
0107                        PACKACK>                ;PACK ACKNOWLEDGE
0113              FUNCTAB +EXE$ONEPARM,-            ;ONE PARAMETER FUNCTIONS
0113                      <SEEK,-                   ;SEEK CYLINDER
0113                        OFFSET,-                ;OFFSET HEADS
0113                        SEARCH>                 ;SEARCH FOR SECTOR
011F              FUNCTAB +EXE$SENSEMODE,-          ;
011F                      <SENSECHAR,-              ;SENSE CHARACTERISTICS
011F                        SENSEMODE>              ;SENSE MODE
012B              FUNCTAB +EXE$SETCHAR,-            ;
012B                      <SETCHAR,-                ;SET CHARACTERISITCS
012B                        SETMODE>                ;SET MODE
```

```
                              0137          .SBTTL  START I/O OPERATION
                              0137   ;+
                              0137   ; DB_STARTIO - START I/O OPERATION ON DEVICE UNIT
                              0137   ;
                              0137   ; THIS ENTRY POINT IS ENTERED TO START AN I/O OPERATION ON A DEVICE UNIT.
                              0137   ;
                              0137   ; INPUTS:
                              0137   ;
                              0137   ;       R3 = ADDRESS OF I/O PACKET.
                              0137   ;       R5 = UCB ADDRESS OF DEVICE UNIT.
                              0137   ;
                              0137   ; OUTPUTS:
                              0137   ;
                              0137   ;       FUNCTION DEPENDENT PARAMETERS ARE STORED IN THE DEVICE UCB, THE ERROR
                              0137   ;       RETRY COUNT IS RESET, AND THE FUNCTION IS EXECUTED. AT FUNCTION COMPLETION
                              0137   ;       THE OPERATION IS TERMINATED THROUGH REQUEST COMPLETE.
                              0137   ;-
                              0137
                              0137   DB_STARTIO:                                     ;START I/O OPERATION
    0080 C5    0081 C5    90  0137          MOVB    UCB$B_ERTMAX(R5),UCB$B_ERTCNT(R5) ;INITIALIZE ERROR RETRY COUNT
                          8A  013E          BICB    #<ERL_M_MEDOFF!-        ; Clear flags used to signal medium
                              013F                  ERL_M_ECC_DEFER>,-      ;  offline and ECC correction deferred
         00D2 C5    05        013F                  UCB$B_DB_ERL(R5)        ;  at start of function.
       009A C5    20 A3   B0  0143          MOVW    IRP$W_FUNC(R3),UCB$W_FUNC(R5) ;SAVE FUNCTION CODE AND MODIFIERS
            50    38 A3   D0  0149          MOVL    IRP$L_MEDIA(R3),R0      ;GET PARAMETER LONGWORD
                              014D
                              014D   ;
                              014D   ; MOVE FUNCTION DEPENDENT PARAMETERS TO UCB
                              014D   ;
                              014D
              06    00    EF  014D   10$:   EXTZV   #IRP$V_FCODE,#IRP$S_FCODE,- ;EXTRACT I/O FUNCTION CODE
          51    20 A3         0150                  IRP$W_FUNC(R3),R1       ;
              51    02    91  0153          CMPB    #IO$_SEEK,R1            ;SEEK FUNCTION?
                    2F    13  0156          BEQL    20$                     ;IF EQL YES
              51    06    91  0158          CMPB    #IO$_OFFSET,R1          ;OFFSET FUNCTION?
                    31    13  015B          BEQL    30$                     ;IF EQL YES
              51    09    91  015D          CMPB    #IO$_SEARCH,R1          ;SEARCH FUNCTION?
                    33    13  0160          BEQL    40$                     ;IF EQL YES
              51    11    91  0162          CMPB    #IO$_AVAILABLE, R1      ;AVAILABLE function?
                    0F    13  0165          BEQL    15$                     ;Branch if yes.
         00BC C5    50    D0  0167          MOVL    R0,UCB$W_DA(R5)         ;STORE PARAMETER LONGWORD
              51    18    91  016C          CMPB    #IO$_WRITECHECKH,R1     ;DISJOINT FUNCTION CODE?
                    29    1A  016F          BGTRU   50$                     ;IF GTRU NO
              51    09    A2  0171          SUBW    #IO$_WRITECHECKH-IO$_READHEAD-1,R1 ;CONVERT TO DENSE FUNCTION CODE
                    24    11  0174          BRB     50$                     ;
                              0176
                              0176   ;
                              0176   ; AVAILABLE FUNCTION - Clear software volume valid bit & exit
                              0176   ;
        64 A5    0800 8F  AA  0176   15$:   BICW    #UCB$M_VALID, UCB$W_STS(R5) ;Clear software volume valid bit.
              50    01    3C  017C          MOVZWL  #SS$_NORMAL, R0         ;Setup success status for zero
              51          D4  017F          CLRL    R1                      ;bytes transfered operation,
                              0181          REQCOM                          ;and complete request.
                              0187
                              0187   ;
                              0187   ; SEEK FUNCTION - SET CYLINDER ADDRESS
                              0187   ;
```

```
                       0187
 00BE C5    50    B0   0187  20$:     MOVW    R0,UCB$W_DC(R5)           ;SET CYLINDER ADDRESS
            0C    11   018C           BRB     50$                       ;
                       018E
                       018E  ;
                       018E  ; OFFSET FUNCTION - SET CURRENT OFFSET VALUE
                       018E  ;
                       018E
 00C8 C5    50    90   018E  30$:     MOVB    R0,UCB$W_OFFSET(R5)       ;SET OFFSET VALUE
            05    11   0193           BRB     50$                       ;
                       0195
                       0195  ;
                       0195  ; SEARCH FUNCTION - SET SECTOR ADDRESS
                       0195  ;
                       0195
 00BC C5    50    90   0195  40$:     MOVB    R0,UCB$W_DA(R5)           ;SET SECTOR ADDRESS
                       019A
                       019A  ;
                       C19A  ; FINISH PREPROCESSING
                       019A  ;
                       019A
 0092 C5    51    90   019A  50$:     MOVB    R1,UCB$B_FEX(R5)          ;SAVE FUNCTION DISPATCH INDEX
    54    24 A5   D0   019F           MOVL    UCB$L_CRB(R5),R4          ;GET ADDRESS OF CRB
    54    2C B4   D0   01A3           MOVL    @CRB$L_INTD+VEC$L_IDB(R4),R4 ;GET FIRST CONTROLLER CSR ADDRESS
00 68 A5    00    E4   01A7           BBSC    #UCB$V_ECC,UCB$W_DEVSTS(R5),FDISPATCH ;CLEAR ECC CORRECTION MADE
                       01AC
                       01AC  ;
                       01AC  ; CENTRAL FUNCTION DISPATCH
                       01AC  ;
                       01AC
                       01AC  FDISPATCH:                                 ;FUNCTION DISPATCH
    53    58 A5   D0   01AC           MOVL    UCB$L_IRP(R5),R3          ;RETRIEVE ADDRESS OF I/O PACKET
0D 2A A3    08    E0   01B0           BBS     #IRP$V_PHYSIO,IRP$W_STS(R3),10$ ;IF SET, PHYSICAL I/O FUNCTION
08 64 A5    0B    E0   01B5           BBS     #UCB$V_VALID,UCB$W_STS(R5),10$ ;IF SET, VOLUME SOFTWARE VALID
  50    0254 8F   3C   01BA           MOVZWL  #SS$_VOLINV,R0            ;SET VOLUME INVALID STATUS
          0582    31   01BF           BRW     RESETXFR                  ;
                       01C2
                       01C2  ;
                       01C2  ; UNIT IS SOFTWARE VALID OR FUNCTION IS PHYSICAL I/O
                       01C2  ;
                       01C2
 50    0092 C5    9A   01C2  10$:     MOVZBL  UCB$B_FEX(R5),R0          ;GET DISPATCH FUNCTION CODE
 00C9 C5    10    90   01C7           MOVB    #RP_OF_M_FMT/256,UCB$W_OFFSET+1(R5) ;CLEAR ECI, HCI, AND SET FORMAT
 00CB C5    01    90   01CC           MOVB    #1,UCB$B_OFFRTC(R5)       ;SET INITIAL OFFSET RETRY COUNT
       00CA C5    94   01D1           CLRB    UCB$B_OFFNDX(R5)          ;CLEAR INITIAL OFFSET TABLE INDEX
                       01D5           CASE    R0,<-                     ;DISPATCH TO FUNCTION HANDLING ROUTINE
                       01D5                   NOP,-                     ;NO OPERATION
                       01D5                   UNLOAD,-                  ;UNLOAD VOLUME
                       01D5                   SEEK,-                    ;SEEK CYLINDER
                       01D5                   RECAL,-                   ;RECALIBRATE
                       01D5                   DRVCLR,-                  ;DRIVE CLEAR
                       01D5                   RELEASE,-                 ;RELEASE PORT
                       01D5                   OFFSET,-                  ;OFFSET HEADS
                       01D5                   RETCENTER,-               ;RETURN HEADS TO CENTER
                       01D5                   PACKACK,-                 ;PACK ACKNOWLEDGE
                       01D5                   SEARCH,-                  ;SEARCH FOR SECTOR
                       01D5                   WRITECHECK,-              ;WRITE CHECK DATA
```

```
                          01D5                WRITEDATA,-             ;WRITE DATA
                          01D5                READDATA,-              ;READ DATA
                          01D5                WRITEHEAD,-             ;WRITE HEADER AND DATA
                          01D5                READHEAD,-              ;READ HEADER AND DATA
                          01D5                WRITECHECKH,-           ;WRITE CHECK HEADER AND DATA
                          01D5                READPRESET-             ;READ IN PRESET
                          01D5                >
                          01FB
                          01FB  ;
                          01FB  ; IO$_UNLOAD INDICATES THE UNIT IS NOT MOUNTED SO WE CLEAR SOFTWARE VOLUME
                          01FB  ; VALID BEFORE EXECUTING THE OPERATION.  IO$_PACKACK INDICATES THAT SOFTWARE
                          01FB  ; IS READY TO MOUNT THE VOLUME SO WE SET SOFTWARE VOLUME VALID BEFORE
                          01FB  ; EXECUTING THE OPERATION.
                          01FB  ;
                          01FB  UNLOAD:
  64 A5    0800 8F    AA  01FB          BICW    #UCB$M_VALID, UCB$W_STS(R5) ;Clear software volume valid bit.
                06    11  0201          BRB     NOP                         ;Proceed with the unload operation.
                          0203
                          0203  PACKACK:
  64 A5    0800 8F    A8  0203          BISW    #UCB$M_VALID, UCB$W_STS(R5) ;Set software volume valid bit.
                          0209  ;       BRB     NOP                         ;Proceed with the unload operation.
                          0209
                          0209  ;
                          0209  ; NO OPERATION, SEEK, RECALIBRATE, DRIVE CLEAR, RELEASE, OFFSET,
                          0209  ; RETURN TO CENTER LINE, SEARCH, AND READ IN PRESET
                          0209  ;
                          0209
                          0209  NOP:                                        ;NO OPERATION
                          0209  SEEK:                                       ;SEEK CYLINDER
                          0209  RECAL:                                      ;RECALIBRATE
                          0209  DRVCLR:                                     ;DRIVE CLEAR
                          0209  RELEASE:                                    ;RELEASE PORT
                          0209  OFFSET:                                     ;OFFSET READ HEADS
                          0209  RETCENTER:                                  ;RETURN TO CENTERLINE
                          0209  SEARCH:                                     ;SEARCH FOR SECTOR
                          0209  READPRESET:                                 ;READIN PRESET
                          0209          EXFUNC  RETRY                       ;EXECUTE HOUSEKEEPING FUNCTION
                73    11  020E          BRB     NORMAL                      ;
                          0210
                          0210  ;
                          0210  ; WRITE CHECK DATA AND WRITE CHECK HEADER AND DATA
                          0210  ;
                          0210
                          0210  WRITECHECK:                                 ;WRITE CHECK DATA
                          0210  WRITECHECKH:                                ;WRITE CHECK HEADER AND DATA
009A C5    4000 8F    AA  0210          BICW    #IO$M_DATACHECK,UCB$W_FUNC(R5) ;CLEAR DATA CHECK REQUEST
                          0217
                          0217  ;
                          0217  ; WRITE DATA, WRITE HEADER AND DATA, WRITE CHECK DATA, AND WRITE CHECK HEADER
                          0217  ; AND DATA
                          0217  ;
                          0217
                          0217  WRITEDATA:                                  ;WRITE DATA
                          0217  WRITEHEAD:                                  ;WRITE HEADER AND DATA
   00C9 C5      08    88  0217          BISB    #RP_OF_M_ECI/256,UCB$W_OFFSET+1(R5) ;INHIBIT ECC CORRECTION
                          021C
                          021C  ;
```

```
                             021C  ; READ DATA, READ HEADER AND DATA, WRITE DATA, WRITE HEADER AND DATA, WRITE
                             021C  ; CHECK DATA, AND WRITE CHECK HEADER AND DATA
                             021C  ;
                             021C
                             021C  READDATA:                                       ;READ DATA
                             021C  READHEAD:                                       ;READ HEADER AND DATA
     08 009A C5    0C    E0  021C          BBS     #IO$V_INHSEEK,UCB$W_FUNC(R5),TRANRQCH ;IF SET, NO EXPLICIT SEEK
                             0222          EXFUNC  RETRY,F_SEARCHA                 ;SEARCH AHEAD OF STARTING SECTOR
                             022A
                             022A  ;
                             022A  ; DATA TRANSFER - REQUEST CHANNEL
                             022A  ;
                             022A
                             022A  TRANRQCH:                                       ;DATA TRANSFER REQUEST CHANNEL
                             022A          REQPCHAN LOW                            ;REQUEST PRIMARY CHANNEL FOR TRANSFER
                             0230
                             0230  ;
                             0230  ; DATA TRANSFER - CHANNEL ALREADY OWNED
                             0230  ;
                             0230
                             0230  TRANNOCH:                                       ;DATA TRANSFER CHANNEL OWNED
        50   0092 C5     9A  0230          MOVZBL  UCB$B_FEX(R5),R0                ;GET FUNCTION DISPATCH INDEX
                             0235          EXFUNC  TRANXT                          ;EXECUTE TRANSFER FUNCTION
                             023A
                             023A  ;
                             023A  ; DATA CHECK
                             023A  ;
                             023A
                             023A  DATACHECK:                                      ;DATA CHECK
     43 009A C5    0E    E1  023A          BBC     #IO$V_DATACHECK,UCB$W_FUNC(R5),NORMAL ;IF CLR, NO DATA CHECK
        50   0639 8F     3C  0240          MOVZWL  #SS$_QASECC,R0                  ;ASSUME ECC CORRECTION WAS MADE
       3C 68 A5    00    E0  0245          BBS     #UCB$V_ECC,UCB$W_DEVSTS(R5),CHECKXT ;IF SET, ECC CORRECTION MADE
                             024A          RELCHAN                                 ;RELEASE CHANNEL
        00C9 C5    19    90  0250          MOVB    #<RP_OF_M_DCK!-                 ;SET DATA CHECK IN PROGRESS,
                             0255                  RP_OF_M_ECI!-                   ;INHIBIT ECC CORRECTION, AND
                             0255                  RP_OF_M_FMT>/256,UCB$W_OFFSET+1(R5) ;SET FORMAT
        00CB C5    01    90  0255          MOVB    #1,UCB$B_OFFRTC(R5)             ;SET INITIAL OFFSET RETRY COUNT
              00CA C5    94  025A          CLRB    UCB$B_OFFNDX(R5)                ;CLEAR INITIAL OFFSET TABLE INDEX
          52    58 A5    D0  025E          MOVL    UCB$L_IRP(R5),R2                ;GET ADDRESS OF IRP
       78 A5    2C A2    7D  0262          MOVQ    IRP$L_SVAPTE(R2),UCB$L_SVAPTE(R5) ;RESET TRANSFER PARAMETERS
     00BC C5    38 A2    D0  0267          MOVL    IRP$L_MEDIA(R2),UCB$W_DA(R5) ;
                             026D
                             026D  ;
                             026D  ; DATA CHECK RETRY
                             026D  ;
                             026D
                             026D  CHECKRETRY:                                     ;DATA CHECK RETRY
                             026D          REQPCHAN LOW                            ;REQUEST PRIMARY CHANNEL FOR DATA CHECK
        50   0092 C5     9A  0273          MOVZBL  UCB$B_FEX(R5),R0                ;GET FUNCTION DISPATCH INDEX
     50   FDB0 CF40      9A  0278          MOVZBL  CHECKTAB-CDF_WRITEDATA[R0],R0 ;GET CASE TABLE INDEX
                             027E          EXFUNC  TRANXT                          ;EXECUTE DATA CHECK FUNCTION
                             0283
                             0283  ;
                             0283  ; SUCCESSFUL OPERATION COMPLETION
                             0283  ;
                             0283
                             0283  NORMAL:                                         ;
```

```
           50    01   3C  0283               MOVZWL  S^#SS$_NORMAL,R0                      ;SET NORMAL COMPLETION STATUS
                          0286      CHECKXT:                                        ;
               01F6   31  0286               BRW     FUNCXT                         ;
                          0289
                          0289      ;
                          0289      ; TRANSFER ENDED WITH A RETRIABLE ERROR
                          0289      ;
                          0289
                          0289      TRANXT:                                         ;TRANSFER EXIT
      0093 C5    0B   91  0289               CMPB    #CDF_WRITEDATA,UCB$B_CEX(R5) ;WRITE DATA FUNCTION?
                 1B   13  028E               BEQL    RETRY                          ;IF EQL YES
      0093 C5    0D   91  0290               CMPB    #CDF_WRITEHEAD,UCB$B_CEX(R5) ;WRITE HEADER FUNCTION?
                 14   13  0295               BEQL    RETRY                          ;IF EQL YES
  51   00064F74 8F    D3  0297               BITL    #MBA$M_SR_DLT!-                ;DATA LATE OR,
                          029E                       MBA$M_SR_INVMAP!-              ;INVALID MAP REGISTER OR,
                          029E                       MBA$M_SR_MAPPE!-               ;MAP REGISTER PARITY ERROR OR,
                          029E                       MBA$M_SR_MCPE!-                ;MASSBUS CONTROL PARITY ERROR OR,
                          029E                       MBA$M_SR_SPE!-                 ;SILO PARITY ERROR OR,
                          029E                       MBA$M_SR_MDPE!-                ;MASSBUS DATA PARITY ERROR OR,
                          029E                       MBA$M_SR_MXF!-                 ;MISSED TRANSFER OR,
                          029E                       MBA$M_SR_NED!-                 ;NONEXISTENT DISK OR,
                          029E                       MBA$M_SR_RDS!-                 ;READ DATA SUBSTITUTE OR,
                          029E                       MBA$M_SR_WCKLWR!-              ;WRITE CHECK LOWER BYTE OR,
                          029E                       MBA$M_SR_WCKUPR,R1             ;WRITE CHECK UPPER BYTE?
                 08   12  029E               BNEQ    RETRY                          ;IF NEQ YES - RETRY FUNCTION
         0A 52   08   E0  02A0               BBS     #RP_ER1_V_HCRC,R2,ECC          ; First check HCRC. If bad go to ECC.
       52   2088 8F   B3  02A4               BITW    #RP_ER1_M_FER!-                ;FORMAT ERROR OR,
                          02A9                       RP_ER1_M_HCE!-                 ; Header Compare Error.
                          02A9                       RP_ER1_M_OPI!-                 ;OPERATION INCOMPLETE OR,
                          02A9                       RP_ER1_M_PAR!-                 ;PARITY ERROR OR,
                          02A9                       RP_ER1_M_WCF,R2                ;WRITE CLOCK FAIL?
                 03   13  02A9               BEQL    ECC                            ;IF EQL NO
                          02AB      RETRY:                                          ;
               0110   31  02AB               BRW     RETRYERR                       ;RETRIABLE ERROR
                          02AE
                          02AE      ;
                          02AE      ; ECC, DRIVE TIMING, OR HEADER ERROR - APPLY ECC OR PERFORM OFFSET RECOVERY
                          02AE      ;
                          02AE
                          02AE      ECC:                                            ;ECC CORRECTION
  51   7E A5   00C0 C5    A1  02AE               ADDW3   UCB$W_BCR(R5), -               ; Compute bytes transfered then
                          02B5                       UCB$W_BCNT(R5), R1             ; clear byte offset bits and
  50   51   FFFF01FF 8F   CB  02B5               BICL3   #^XFFFF01FF, R1, R0            ; convert result to a longword.
                 6D   13  02BD               BEQL    OFF                            ; Branch if whole blocks xfered is zero.
       51   01FF 8F   B3  02BF               BITW    #^X1FF, R1                     ; Was a partial block transfered?
                 66   12  02C4               BNEQ    OFF                            ; Branch if partial block transfered.
       52   0180 8F   B3  02C6               BITW    #RP_ER1_M_HCE!-                ; Was there an error while processing
                          02CB                       RP_ER1_M_HCRC, R2              ; the header?
                 07   12  02CB               BNEQ    10$                            ; Branch if header error.
     50   00000200 8F   C2  02CD               SUBL2   #512, R0                       ; Else, reduce bytes xfered by a block.
       52   11C0 8F   B3  02D4      10$:     BITW    #RP_ER1_M_DTE!-                ; For:  DRIVE TIMING ERROR
                          02D9                       RP_ER1_M_ECH!-                 ;       ECC HARD ERROR
                          02D9                       RP_ER1_M_HCE!-                 ;       HEADER COMPARE ERROR
                          02D9                       RP_ER1_M_HCRC,R2               ;       HEADER CRC ERROR
                 51   12  02D9               BNEQ    OFF                            ;          perform offset recovery.
    48 00C8 C5   0B   E0  02DB               BBS     #RP_OF_V_ECI,UCB$W_OFFSET(R5),OFF  ; Branch if ECC inhibited.
            7E   52   7D  02E1               MOVQ    R2,-(SP)                       ; Save work registers.
```

```
 52   00C6 C5    0B    00    EA  02E4          FFS     #0,#11,UCB$W_EC2(R5),R2 ; Find the first error bit in the ECC
                                 02EB                                          ;  pattern.
             53    0A    52    C3  02EB          SUBL3   R2,#10,R3               ; Get the number of error bits
                                 02EF                                          ;  remaining in the pattern.
                         09    15  02EF          BLEQ    20$                     ; Branch if no other bits in pattern.
                         52    D6  02F1          INCL    R2                      ; Point at next bit in pattern.
 52   00C6 C5    53    52    EF  02F3          EXTZV   R2,R3,UCB$W_EC2(R5),R2  ; Is there more than one error bit set?
                         0C    BA  02FA  20$:    POPR    #^M<R3,R2>              ; Restore work registers without
                                 02FC                                          ;  affecting flags.
                         29    1A  02FC          BGTRU   DEFER_ECC               ; If more than one error bit set, don't
                                 02FE                                          ;  apply ECC correction.
                                 02FE  ;
                                 02FE  ; APPLY_ECC -
                                 02FE  ;
                                 02FE  ; Apply ECC correction to correct a single bit error.
                                 02FE  ;
                                 02FE
                                 02FE  APPLY_ECC:
                 7E    51    3C  02FE          MOVZWL  R1, -(SP)               ; Save total bytes transfered, inc. ECC.
            00000000'GF    16  0301          JSB     G^IOC$APPLYECC          ;APPLY ECC CORRECTION
                       50 8ED0  0307          POPL    R0                      ;RETRIEVE TRANSFERED BYTE COUNT
            00000000'GF    16  030A          JSB     G^IOC$UPDATRANSP        ;UPDATE TRANSFER PARAMETERS
                  00CA C5    94  0310          CLRB    UCB$B_OFFNDX(R5)        ; Reset offset table index.
                                 0314          EXFUNC  FATALERR,F_RETCENTER    ; Return to centerline.
                    7E A5    B5  031C          TSTW    UCB$W_BCNT(R5)          ;ANY MORE TO TRANSFER?
                       03    13  031F          BEQL    20$                     ;IF EQL NO
                     FF0C    31  0321          BRW     TRANNOCH                ;TRANSFER NEXT SEGMENT
                     FF13    31  0324  20$:    BRW     DATACHECK               ;CHECK FOR WRITE CHECK
                                 0327
                                 0327  ;
                                 0327  ; DEFER_ECC -
                                 0327  ;
                                 0327  ; Don't apply ECC correction for multiple bit errors unless the error cannot
                                 0327  ; be recovered with offset retries.
                                 0327  ;
                                 0327
                                 0327  DEFER_ECC:
                       04    88  0327          BISB    #ERL_M_ECC_DEFER,-      ; Set flag to indicate that ECC
                  00D2 C5        0329                  UCB$B_DB_ERL(R5)        ;  can be used if offset recovery fails.
                                 032C
                                 032C  ;
                                 032C  ; OFF - OFFSET RECOVERY
                                 032C  ;
                                 032C  ; THIS CODE IS EXECUTED WHEN A DRIVE TIMING ERROR, HEADER COMPARE, OR ECC
                                 032C  ; HARD ERROR IS DETECTED ON A READ FUNCTION.
                                 032C  ;
                                 032C
                                 032C  OFF:                                            ;OFFSET RECOVERY
                       50    D5  032C          TSTL    R0                      ;ANY GOOD DATA TRANSFERED?
                       2E    13  032E          BEQL    20$                     ;IF EQL NO
                                 0330
                                 0330  ;
                                 0330  ; THE TRANSFER ENDED IN AN ERROR BUT THERE WERE SECTORS TRANSFERED THAT
                                 0330  ; CONTAINED GOOD DATA. SINCE THE ERROR COULD HAVE BEEN CAUSED BY A CYLIN-
                                 0330  ; DER CROSSING, THE GOOD DATA IS SAVED AND THE TRANSFER IS RETRIED FROM THE
                                 0330  ; POINT OF ERROR.
                                 0330  ;
```

```
                            0330    849
      00000000'GF     16    0330    850             JSB     G^IOC$UPDATRANSP          ;UPDATE TRANSFER PARAMETERS
           00CA C5    94    0336    851             CLRB    UCB$B_OFFNDX(R5)          ;RESET OFFSET TABLE INDEX
      00CB C5   10    90    033A    852     10$:    MOVB    #16,UCB$B_OFFRTC(R5)      ;SET OFFSET RETRY COUNT
      00CA C5   08    91    033F    853             CMPB    #OFFSIZ,UCB$B_OFFNDX(R5)  ;ALL OFFSETS TRIED?
                08    12    0344    854             BNEQ    15$                       ; Branch if not.
                02    E4    0346    855             BBSC    #ERL_V_ECC_DEFER,-        ; Correct the error with ECC if we can.
           00D2 C5          0348    856                     UCB$B_DB_ERL(R5),-        ;
                B2          034B    857                     APPLY_ECC                 ;
                65    11    034C    858             BRB     OFFSETERR                 ; Otherwise, fatal error.
                            034E    859     15$:    RELCHAN                           ;RELEASE CHANNEL
                            0354    860             EXFUNC  FATALERR,F_RETCENTER      ;RETURN TO CENTERLINE
                44    11    035C    861             BRB     50$                       ;
                            035E    862
                            035E    863     ;
                            035E    864     ; NO GOOD DATA TRANSFERED - CHECK IF CHANGE IN OFFSET NEEDED
                            035E    865     ;
                            035E    866
   52    90C0 8F      B3    035E    867     20$:    BITW    #RP_ER1_M_DCK!-           ;DATA CHECK OR,
                            0363    868                     RP_ER1_M_DTE!-            ;DRIVE TIMING OR,
                            0363    869                     RP_ER1_M_ECH!-            ;ECC HARD ERROR OR,
                            0363    870                     RP_ER1_M_HCE,R2           ;HEADER COMPARE ERROR?
                05    12    0363    871             BNEQ    30$                       ;IF NEQ YES
      00C9 C5   04    88    0365    872             BISB    #RP_OF_M_HCI/256,UCB$W_OFFSET+1(R5) ;SET HEADER COMPARE INHIBIT
           00CB C5    97    036A    873     30$:    DECB    UCB$B_OFFRTC(R5)          ;CHANGE CURRENT OFFSET?
                37    12    036E    874             BNEQ    60$                       ;IF NEQ NO
           00CA C5    96    0370    875             INCB    UCB$B_OFFNDX(R5)          ;UPDATE OFFSET TABLE INDEX
      50   00CA C5    9A    0374    876             MOVZBL  UCB$B_OFFNDX(R5),R0       ;GET NEXT OFFSET TABLE INDEX
   50    FD1C CF40    9A    0379    877             MOVZBL  OFFTAB-1[R0],R0           ;GET NEXT OFFSET VALUE?
                B9    13    037F    878             BEQL    10$                       ;IF EQL RETURN TO CENTERLINE
         18 A3  02    D3    0381    879             BITL    #2,RP_DT(R3)              ;RP06 DRIVE?
                03    12    0385    880             BNEQ    40$                       ;IF NEQ YES
           50   02    C4    0387    881             MULL    #2,R0                     ;CONVERT TO RP04 OFFSET VALUE
      00C8 C5   50    90    038A    882     40$:    MOVB    R0,UCB$W_OFFSET(R5)       ;SET NEW OFFSET VALUE
      00CB C5   02    90    038F    883             MOVB    #2,UCB$B_OFFRTC(R5)       ;SET OFFSET RETRY COUNT
                            0394    884             RELCHAN                           ;RELEASE CHANNEL
                            039A    885             EXFUNC  FATALERR,F_OFFSET         ;OFFSET TO NEXT POSITION
      00C9 C5   04    8A    03A2    886     50$:    BICB    #RP_OF_M_HCI/256,UCB$W_OFFSET+1(R5) ;CLEAR HEADER COMPARE INHIBIT
   03 00C8 C5   08    E0    03A7    887     60$:    BBS     #RP_OF_V_DCK,UCB$W_OFFSET(R5),70$ ;IF SET, DATA CHECK FUNCTION
              FE7A    31    03AD    888             BRW     TRANRQCH                  ;TRY FUNCTION AGAIN
              FEBA    31    03B0    889     70$:    BRW     CHECKRETRY                ;TRY DATA CHECK AGAIN
                            03B3    890
                            03B3    891     ;
                            03B3    892     ; ALL OFFSETS TRIED - RETRIEVE FINAL TRANSFER STATUS
                            03B3    893     ;
                            03B3    894
                            03B3    895     OFFSETERR:                                ;OFFSET RECOVERY ERROR
         50   04 A3   D0    03B3    896             MOVL    RP_DS(R3),R0              ;RETRIEVE FINAL DRIVE STATUS
      51   00CE C5    D0    03B7    897             MOVL    UCB$L_DB_SR(R5),R1        ;RETRIEVE FINAL ERROR STATUS
                2D    11    03BC    898             BRB     FATALERR                  ; Branch around.
                            03BE    899
                            03BE    900     ;
                            03BE    901     ; RETRIABLE ERROR
                            03BE    902     ;
                            03BE    903
                            03BE    904     RETRYERR:                                 ;RETRIABLE ERROR
                07    BB    03BE    905             PUSHR   #^M<R0,R1,R2>             ; Save volital error status registers.
```

```
                      03C0   906             RELCHAN                           ; Release channel before possible RECAL
           07    BA   03C6   907             POPR     #^M<R0,R1,R2>            ; Restore error status registers.
                      03C8   908
                      03C8   909  ;
                      03C8   910  ; Here we will do a RECAL if we had either a Seek Incomplete or a Header
                      03C8   911  ;          Compare Error.
                      03C8   912  ;
                      03C8   913
           0E    E0   03C8   914             BBS      #RP_ER3_V_SKI,-          ; If Seek Incomplete
    04 00CC C5        03CA   915                      UCB$W_DB_ER3(R5),10$     ;   then go do RECAL.
    08 52   07   E1   03CE   916             BBC      #RP_ER1_V_HCE,R2,20$     ; If NOT HCE, then branch around RECAL.
                      03D2   917 10$:        EXFUNC   FATALERR,F_RECAL         ; Do RECAL for SKI or HCE.
       0080 C5   97   03DA   918 20$:        DECB     UCB$B_ERTCNT(R5)         ;ANY RETRIES LEFT?
           08    13   03DE   919             BEQL     FATALERR                 ;IF EQL NO
                      03E0   920             EXFUNC   FATALERR,F_DRVCLR        ; Issue drive clear before retrying.
         FDC1    31   03E8   921             BRW      FDISPATCH                ;
                      03EB   922
                      03EB   923  ;
                      03EB   924  ; FATAL CONTROLLER/DRIVE ERROR, ERROR RETRY COUNT EXHAUSTED, ERROR RETRY
                      03EB   925  ; INHIBITED, OR FINAL OFFSET TRIED
                      03EB   926  ;
                      03EB   927
                      03EB   928 FATALERR:                                     ;FATAL ERROR - SET STATUS
    0E 50   0C   E0   03EB   929             BBS      #RP_DS_V_MOL,R0,10$      ; Branch if medium is online.
 50   01A4 8F    3C   03EF   930             MOVZWL   #SS$_MEDOFL,R0           ; Otherwise, set medium offline status,
      0800 8F    AA   03F4   931             BICW     #UCB$M_VALID,-           ;   clear software volume valid,
           64 A5      03F8   932                      UCB$W_STS(R5)            ;
            0082 31   03FA   933             BRW      FUNCXT                   ;   and branch to common exit.
    79 50   06   E1   03FD   934 10$:        BBC      #RP_DS_V_VV,R0,20$       ;IF CLR, VOLUME INVALID
 50   023C 8F    3C   0401   935             MOVZWL   #SS$_UNSAFE,R0           ;SET DRIVE UNSAFE STATUS
    75 52   0E   E0   0406   936             BBS      #RP_ER1_V_UNS,R2,FUNCXT  ;IF SET, DRIVE UNSAFE
 50   02D4 8F    3C   040A   937             MOVZWL   #SS$_OPINCOMPL,R0        ;SET OPERATION INCOMPLETE STATUS
    6C 52   0D   E0   040F   938             BBS      #RP_ER1_V_OPI,R2,FUNCXT  ;IF SET, OPERATION INCOMPLETE
 50   00BC 8F    3C   0413   939             MOVZWL   #SS$_FORMAT,R0           ;SET FORMAT ERROR STATUS
    63 52   04   E0   0418   940             BBS      #RP_ER1_V_FER,R2,FUNCXT  ;IF SET, FORMAT ERROR
 50   025C 8F    3C   041C   941             MOVZWL   #SS$_WRITLCK,R0          ;SET WRITE LOCK ERROR STATUS
    5A 52   0B   E0   0421   942             BBS      #RP_ER1_V_WLE,R2,FUNCXT  ;IF SET, WRITE LOCK ERROR
 50   0134 8F    3C   0425   943             MOVZWL   #SS$_IVADDR,R0           ;SET INVALID DISK ADDRESS STATUS
 52   0600 8F    B3   042A   944             BITW     #RP_ER1_M_AOE!-          ;DISK ADDRESS OVERFLOW OR,
                      042F   945                      RP_ER1_M_IAE,R2          ;INVALID DISK ADDRESS ERROR?
           4E    12   042F   946             BNEQ     FUNCXT                   ;IF NEQ YES
 50   008C 8F    3C   0431   947             MOVZWL   #SS$_DRVERR,R0           ;SET DRIVE ERROR STATUS
 52   1027 8F    B3   0436   948             BITW     #RP_ER1_M_DTE!-          ;DRIVE TIMING ERROR OR,
                      043B   949                      RP_ER1_M_ILF!-           ;ILLEGAL FUNCTION OR,
                      043B   950                      RP_ER1_M_ILR!-           ;ILLEGAL REGISTER OR,
                      043B   951                      RP_ER1_M_RMR!-           ;REGISTER MODIFY REFUSE OR,
                      043B   952                      RP_ER1_M_WCF,R2          ;WRITE CLOCK FAIL ERROR?
           42    12   043B   953             BNEQ     FUNCXT                   ;IF NEQ YES
 50   01F4 8F    3C   043D   954             MOVZWL   #SS$_PARITY,R0           ; Set parity error status.
 52   8140 8F    B3   0442   955             BITW     #RP_ER1_M_DCK!-          ; Data check error or,
                      0447   956                      RP_ER1_M_ECH!-           ;   ECC hard error or,
                      0447   957                      RP_ER1_M_HCRC,R2         ;   header CRC error?
           36    12   0447   958             BNEQ     FUNCXT                   ; Branch if so.
 50   0054 8F    3C   0449   959             MOVZWL   #SS$_CTRLERR,R0          ; Set fatal controller error status.
 52   0088 8F    B3   044E   960             BITW     #RP_ER1_M_HCE!-          ; Header compare error or,
                      0453   961                      RP_ER1_M_PAR,R2          ;   parity error?
           2A    12   0453   962             BNEQ     FUNCXT                   ; Branch if so.
```

```
          51   00024064 8F    D3  0455          BITL    #MBA$M_SR_MAPPE!-          ;MAP PARITY ERROR OR,
                                  045C                  MBA$M_SR_MCPE!-            ;MASSBUS CONTROL PARITY ERROR OR,
                                  045C                  MBA$M_SR_SPE!-             ;SILO PARITY ERROR OR,
                                  045C                  MBA$M_SR_MDPE!-            ;MASSBUS DATA PARITY ERROR OR,
                                  045C                  MBA$M_SR_RDS,R1            ;READ DATA SUBSTITUTE?
                          21  12  045C          BNEQ    FUNCXT                     ;IF NEQ YES
               50   005C  8F  3C  045E          MOVZWL  #SS$_DATACHECK,R0          ;SET DATA CHECK ERROR STATUS
               51   0600  8F  B3  0463          BITW    #MBA$M_SR_WCKLWR!-         ;WRITE CHECK ERROR LOWER BYTE OR,
                                  0468                  MBA$M_SR_QCKUPR,R1         ;WRITE CHECK ERROR UPPER BYTE?
                          15  12  0468          BNEQ    FUNCXT                     ;IF NEQ YES
               50   01C4  8F  3C  046A          MOVZWL  #SS$_NONEXDRV,R0           ;SET NONEXISTENT DRIVE STATUS
                  0C  51  12  E0  046F          BBS     #MBA$V_SR_NED,R1,FUNCXT    ;IF SET, NONEXISTENT DRIVE
               50   0054  8F  3C  0473          MOVZWL  #SS$_CTRLERR,R0            ;SET CONTROLLER ERROR STATUS
                          05  11  0478          BRB     FUNCXT                     ;
               50   0254  8F  3C  047A  20$:    MOVZWL  #SS$_VOLINV,R0             ;SET VOLUME INVALID STATUS
                                  047F
                                  047F  ;
                                  047F  ; FUNCTION COMPLETION COMMON EXIT
                                  047F  ;
                                  047F
                                  047F  FUNCXT:                                    ;FUNCTION EXIT
                          50  DD  047F          PUSHL   R0                         ;SAVE FINAL REQUEST STATUS
               00000000'GF    16  0481          JSB     G^IOC$DIAGBUFILL           ;FILL DIAGNOSTIC BUFFER IF PRESENT
                                  0487          RELCHAN                            ;RELEASE CHANNEL IF OWNED
               0092 C5    0A  91  048D          CMPB    #CDF_WRITECHECK,UCB$B_FEX(R5) ;DRIVE RELATED FUNCTION?
                          13  1A  0492          BGTRU   10$                        ;IF GTRU YES
               0092 C5    10  91  0494          CMPB    #CDF_READPRESET,UCB$B_FEX(R5) ;READIN PRESET FUNCTION?
                          0C  13  0499          BEQL    10$                        ;IF EQL YES
                 53   58  A5  D0  049B          MOVL    UCB$L_IRP(R5),R3           ;RETRIEVE ADDRESS OF IRP
    02 AE    32 A3  00C0  C5  A1  049F          ADDW3   UCB$W_BCR(R5),IRP$W_BCNT(R3),2(SP) ;CALCULATE BYTES TRANSFERED
                          51  D4  04A7  10$:    CLRL    R1                         ;CLEAR SECOND STATUS LONGWORD
                          50 8ED0 04A9          POPL    R0                         ;RETRIEVE FINAL REQUEST STATUS
              53    0091  C5  9A  04AC          MOVZBL  UCB$B_SLAVE+1(R5),R3       ;Get drive offset constant
            53    0400 C443   DE  04B1          MOVAL   MBA$L_ERB(R4)[R3],R3       ;Get address of driver registers
                    63    09  9A  04B7          MOVZBL  #F_DRVCLR!1,RP_CS1(R3)     ;Issue drive clear before release
                    63    0B  9A  04BA          MOVZBL  #F_RELEASE!1,RP_CS1(R3)    ;Release port
                                  04BD          REQCOM                             ;COMPLETE REQUEST
```

```
04C3          .SBTTL  RP04/05/06 HARDWARE FUNCTION EXECUTION
04C3  ;
04C3  ; FEX - RP04/05/06 HARDWARE FUNCTION EXECUTION
04C3  ;
04C3  ; THIS ROUTINE IS CALLED VIA A BSB WITH A BYTE IMMEDIATELY FOLLOWING THAT
04C3  ; SPECIFIES THE ADDRESS OF AN ERROR ROUTINE. ALL DATA IS ASSUMED TO HAVE BEEN
04C3  ; SET UP IN THE UCB BEFORE THE CALL. THE APPROPRIATE PARAMETERS ARE LOADED
04C3  ; INTO DEVICE REGISTERS AND THE FUNCTION IS INITIATED. IF THE FUNCTION IS AN
04C3  ; IMMEDIATE FUNCTION CONTROL RETURNS IMMEDIATELY. ELSE THE RETURN ADDRESS
04C3  ; IS STORED IN THE UCB AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
04C3  ; RUPT OCCURS, CONTROL IS RETURNED TO THE CALLER.
04C3  ;
04C3  ; INPUTS:
04C3  ;
04C3  ;       R0 = FUNCTION TABLE DISPATCH INDEX.
04C3  ;       R3 = ADDRESS OF DRIVE CONTROL STATUS REGISTER 1.
04C3  ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
04C3  ;       R5 = DEVICE UNIT UCB ADDRESS.
04C3  ;
04C3  ;       00(SP) = RETURN ADDRESS OF CALLER.
04C3  ;       04(SP) = RETURN ADDRESS OF CALLER'S CALLER.
04C3  ;
04C3  ;       IMMEDIATELY FOLLOWING INLINE AT THE CALL SITE IS A BYTE WHICH CONTAINS
04C3  ;       A BRANCH DESTINATION TO AN ERROR RETRY ROUTINE.
04C3  ;
04C3  ; OUTPUTS:
04C3  ;
04C3  ;       THERE ARE FOUR EXITS FROM THIS ROUTINE:
04C3  ;
04C3  ;       1. SPECIAL CONDITION - THIS EXIT IS TAKEN IF A POWER FAILURE OCCURS
04C3  ;               OR THE OPERATION TIMES OUT. IT IS A JUMP TO THE APPROPRIATE
04C3  ;               ERROR ROUTINE.
04C3  ;
04C3  ;       2. FATAL ERROR - THIS EXIT IS TAKEN IF A FATAL CONTROLLER OR DRIVE
04C3  ;               ERROR OCCURS OR IF ANY ERROR OCCURS AND ERROR RETRY IS
04C3  ;               INHIBITED. IT IS A JUMP TO THE FATAL ERROR EXIT ROUTINE.
04C3  ;
04C3  ;       3. RETRIABLE ERROR - THIS EXIT IS TAKEN IF A RETRIABLE CONTROLLER
04C3  ;               OR DRIVE ERROR OCCURS AND ERROR RETRY IS NOT INHIBITED.
04C3  ;               IT CONSISTS OF TAKING THE ERROR BRANCH EXIT.
04C3  ;
04C3  ;       4. SUCCESSFUL OPERATION - THIS EXIT IS TAKEN IF NO ERROR OCCURS
04C3  ;               DURING THE OPERATION. IT CONSISTS OF A RETURN INLINE.
04C3  ;
04C3  ;       IN ALL CASES IF AN ERROR OCCURS, AN ATTEMPT IS MADE TO LOG THE ERROR.
04C3  ;
04C3  ;       IN ALL CASES FINAL DRIVE AND CONTROLLER REGISTERS ARE RETURNED VIA
04C3  ;       THE GENERAL REGISTERS R0, R1, AND R2, AND THE UCB.
04C3  ;
04C3  ;       R0 = DRIVE STATUS REGISTER.
04C3  ;       R1 = MBA STATUS REGISTER.
04C3  ;       R2 = DRIVE ERROR REGISTER 1.
04C3  ;
04C3  ;       UCB$W_EC1(R5) = ECC POSITION REGISTER.
04C3  ;       UCB$W_EC2(R5) = ECC PATTERN REGISTER.
04C3  ;       UCB$W_BCR(R5) = BYTE COUNT REGISTER.
04C3  ;
```

```
                          04C3  1058
                          04C3  1059 FEX:                                           ;FUNCTION EXECUTOR
           009C C5 8ED0   04C3  1060          POPL    UCB$L_DPC(R5)                 ;SAVE DRIVER PC VALUE
       0093 C5   50   90  04C8  1061          MOVB    R0,UCB$B_CEX(R5)              ;SAVE CASE INDEX
        53   0091 C5  9A  04CD  1062          MOVZBL  UCB$B_SLAVE+1(R5),R3          ;GET DRIVE OFFSET CONSTANT
       53  0400 C443  DE  04D2  1063          MOVAL   MBA$L_ERB(R4)[R3],R3          ;GET ADDRESS OF DRIVE REGISTERS
  38 A5   00008000 8F D3  04D8  1064          BITL    #DEV$M_DUA,UCB$L_DEVCHAR(R5)  ;DUAL PORTED DRIVE?
                  40  12  04E0  1065          BNEQ    SEIZE                         ;IF NEQ, YES
        50   0093 C5  9A  04E2  1066 GO:      MOVZBL  UCB$B_CEX(R5),R0              ;RESTORE CASE INDEX (FUNC. CODE)
                          04E7  1067          CASE    R0,<-                         ;DISPATCH TO PROPER FUNCTION ROUTINE
                          04E7  1068                  POSIT,-                       ;SEEK CYLINDER
                          04E7  1069                  EXFNC,-                       ;RECALIBRATE
                          04E7  1070                  IMMED,-                       ;DRIVE CLEAR
                          04E7  1071                  IMMED,-                       ;RELEASE DRIVE
                          04E7  1072                  EXFNC,-                       ;OFFSET HEADS
                          04E7  1073                  EXFNC,-                       ;RETURN TO CENTERLINE
                          04E7  1074                  IMMED,-                       ;PACK ACKNOWLEDGE
                          04E7  1075                  POSIT,-                       ;SEARCH FOR SECTOR
                          04E7  1076                  XFER,-                        ;WRITE CHECK
                          04E7  1077                  XFER,-                        ;WRITE DATA
                          04E7  1078                  XFER,-                        ;READ DATA
                          04E7  1079                  XFER,-                        ;WRITE HEADER AND DATA
                          04E7  1080                  XFER,-                        ;READ HEADER AND DATA
                          04E7  1081                  XFER,-                        ;WRITE CHECK HEADER AND DATA
                          04E7  1082                  IMMED,-                       ;READIN PRESET
                          04E7  1083                  SEARCHA,-                     ;SEARCH AHEAD FOR SECTOR
                          04E7  1084                  >,LIMIT=#CDF_SEEK             ;
                          050B  1085
                          050B  1086 ;
                          050B  1087 ; IMMEDIATE FUNCTION EXECUTION
                          050B  1088 ;
                          050B  1089 ;        FUNCTIONS INCLUDE:
                          050B  1090 ;
                          050B  1091 ;                NO OPERATION,
                          050B  1092 ;                UNLOAD VOLUME,
                          050B  1093 ;                DRIVE CLEAR,
                          050B  1094 ;                RELEASE PORT,
                          050B  1095 ;                READ IN PRESET, AND
                          050B  1096 ;                PACK ACKNOWLEDGE.
                          050B  1097 ;
                          050B  1098 ; THESE FUNCTIONS ARE EXECUTED IMMEDIATELY AND THE FINAL DEVICE REGISTERS
                          050B  1099 ; ARE RETURNED TO THE CALLER.
                          050B  1100 ;
                          050B  1101
                          050B  1102 IMMED:                                         ;IMMEDIATE FUNCTION EXECUTION
                          050B  1103          DSBINT                                ;DISABLE INTERRUPTS
      09 64 A5   05   E0  0511  1104          BBS     #UCB$V_POWER,UCB$W_STS(R5),10$ ;IF SET, POWER HAS FAILED
            63   09   9A  0516  1105          MOVZBL  #F_DRVCLR!1,RP_CS1(R3)        ;CLEAR DRIVE ERRORS
     63   FB6B CF40   9A  0519  1106          MOVZBL  FTAB[R0],RP_CST(R3)           ;EXECUTE FUNCTION
                0082  31  051F  1107 10$:     BRW     ENBXIT                        ;
                          0522  1108
                          0522  1109 ;
                          0522  1110 ; ATTEMPT TO SEIZE THE PORT ON A DUAL PORTED DISK.
                          0522  1111 ;
                          0522  1112
     BA 00D2 C5   01  E1  0522  1113 SEIZE:   BBC     #ERL_V_DUALPORT, -            ;IF CLEAR, NO DUAL PORT KIT IN DRIVE
                          0528  1114                  UCB$B_DB_ERL(R5),GO
                          0528                DSBINT                                  ;DISABLE INTERRUPTS
             04 A3    D4  052E                CLRL    RP_DS(R3)                       ;ATTEMPT TO SEIZE PORT
       00000100 BF    D3  0531                BITL    #RP_DS_M_DPR,-                  ;DID WE SEIZED THE PORT?
             04 A3        0537                        RP_DS(R3)
                12    12  0539                BNEQ    2$                              ;IF NEQ, WE SEIZED THE PORT
                          053B                WFIKPCH RETREG,#15                      ;LETS WAIT FOR THE PORT, ELSE TIMEOUT
                          0545                IOFORK                                  ;CREATE FORK PROCESS
                95    11  054B                BRB     GO                              ;LETS CONTINUE WE HAVE THE PORT
                          054D   2$:          ENBINT                                  ;ENABLE INTERRUPTS
                90    11  0550                BRB     GO                              ;LETS CONTINUE WE HAVE THE PORT
                          0552
                          0552   ; SEARCH AHEAD FUNCTION EXECUTION
                          0552   ;
                          0552   ; THIS FUNCTION MINIMIZES ROTATIONAL LATENCY BY SEARCHING FOR THE SECTOR THAT IS
                          0552   ; FOUR SECTORS AHEAD OF THE STARTING SECTOR OF A TRANSFER.
                          0552   ;
                          0552   ; THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS ARE LOADED, THE
                          0552   ; FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
                          0552   ; RUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                          0552   ;
                          0552
                          0552   SEARCHA:                                             ;
    51    00BC C5     3C  0552                MOVZWL  UCB$W_DA(R5),R1                 ;GET DESIRED TRACK AND SECTOR ADDRESS
       51    04       82  0557                SUBB    #4,R1                           ;COMPUTE FOUR SECTORS BEFORE SPECIFIED SECTO
             04       18  055A                BGEQ    10$                             ;IF GEQ BEFORE SECTOR ZERO
    51    44 A5       80  055C                ADDB    UCB$B_SECTORS(R5),R1            ;CONVERT TO AFTER SECTOR ZERO
    14 A3    51       D0  0560   10$:         MOVL    R1,RP_DA(R3)                    ;SET TRACK AND SECTOR ADDRESS
             15       11  0564                BRB     LDCYL                           ;
                          0566
                          0566   ;
                          0566   ; TRANSFER FUNCTION EXECUTION
                          0566   ;
                          0566   ;        FUNCTIONS INCLUDE:
                          0566   ;
                          0566   ;                 WRITE CHECK,
                          0566   ;                 WRITE CHECK HEADER AND DATA,
                          0566   ;                 WRITE DATA,
                          0566   ;                 WRITE HEADER AND DATA,
                          0566   ;                 READ DATA, AND
                          0566   ;                 READ HEADER AND DATA.
                          0566   ;
                          0566   ; THE MAP REGISTERS, BYTE COUNT REGISTER, AND VIRTUAL ADDRESS REGISTER ARE
                          0566   ; LOADED FOLLOWED BY THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS.
                          0566   ; THE FUNCTION IS INITIATED AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE
                          0566   ; INTERRUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                          0566   ;
                          0566   ; IT ASSUMED THAT THE CALLER OWNS THE CHANNEL ON WHICH THE I/O IS TO OCCUR.
                          0566   ;
                          0566
                          0566   XFER:                                                ;TRANSFER FUNCTION EXECUTION
       08 A4    00    D2  0566                MCOML   #0,MBA$L_SR(R4)                 ;CLEAR MASSBUS ADAPTER ERRORS
                          056A                LOADMBA                                 ;LOAD MAP, BYTE COUNT, AND VIRTUAL ADDRESS
    50    0093 C5     9A  0570                MOVZBL  UCB$B_CEX(R5),R0                ;RETRIEVE FUNCTION TABLE INDEX
                          0575
                          0575   ;
                          0575   ; POSITIONING FUNCTION EXECUTION
                          0575   ;
```

```
                                 0575  ;       FUNCTIONS INCLUDE:
                                 0575  ;
                                 0575  ;               SEEK CYLINDER, AND
                                 0575  ;               SEARCH FOR SECTOR.
                                 0575  ;
                                 0575  ; THE DESIRED CYLINDER, TRACK, AND SECTOR ADDRESS REGISTERS ARE LOADED, THE
                                 0575  ; FUNCTION IS INITIATED, AND A WAITFOR INTERRUPT IS EXECUTED. WHEN THE INTER-
                                 0575  ; RUPT OCCURS, THE FINAL DEVICE REGISTERS ARE RETURNED TO THE CALLER.
                                 0575  ;
                                 0575
                                 0575 POSIT:                                           ;POSITION FUNCTION EXECUTION
        14 A3   00BC C5     3C   0575         MOVZWL  UCB$W_DA(R5),RP_DA(R3)   ;SET DESIRED TRACK AND SECTOR ADDRESS
                                 057B LDCYL:                                           ;
        28 A3   00BE C5     3C   057B         MOVZWL  UCB$W_DC(R5),RP_DC(R3)   ;SET DESIRED CYLINDER ADDRESS
                                 0581
                                 0581  ;
                                 0581  ; INTERRUPT WAIT FUNCTION EXECUTION
                                 0581  ;
                                 0581  ;       FUNCTIONS INCLUDE:
                                 0581  ;
                                 0581  ;               OFFSET HEADS,
                                 0581  ;               RECALIBRATE, AND
                                 0581  ;               RETURN TO CENTERLINE.
                                 0581  ;
                                 0581  ; THE OFFSET REGISTER IS LOADED, THE FUNCTION IS INITIATED, AND A WAITFOR
                                 0581  ; INTERRUPT IS EXECUTED. WHEN THE INTERRUPT OCCURS, THE FINAL DEVICE REGISTERS
                                 0581  ; ARE RETURNED TO THE CALLER.
                                 0581  ;
                                 0581
                                 0581 EXFNC:                                           ;EXECUTE FUNCTION
                                 0581         DSBINT                                   ;DISABLE INTERRUPTS
     48 64 A5      05       E0   0587         BBS     #UCB$V_POWER,UCB$W_STS(R5),ENBXIT ;IF SET, POWER FAILED
           63      09       9A   058C         MOVZBL  #F_DRVCLR!1,RP_CS1(R3)   ;CLEAR DRIVE ERRORS
        24 A3   00C8 C5     3C   058F         MOVZWL  UCB$W_OFFSET(R5),RP_OF(R3) ;SET FORMAT, INHIBIT BITS, AND OFFSET
        52    04 A3  13     78   0595         ASHL    #31-RP_DS_V_MOL,RP_DS(R3),R2 ;MEDIUM ONLINE?
                     1E     18   059A         BGEQ    10$                      ;IF GEQ NO
        63    FAE8 CF40     9A   059C         MOVZBL  FTAB[R0],RP_CS1(R3)      ;INITIATE FUNCTION
                                 05A2         WFIKPCH RETREG,#15               ;WAITFOR INTERRUPT AND KEEP CHANNEL
      00CE C5     08 A4     D0   05AC         MOVL    MBA$L_SR(R4),UCB$L_DB_SR(R5) ;SAVE FINAL CONTROLLER STATUS
                                 05B2         IOFORK                           ;CREATE FORK PROCESS
                     1D     11   05B8         BRB     RETREG                   ;
                                 05BA
                                 05BA  ;
                                 05BA  ; MEDIUM OFFLINE AT START OF FUNCTION
                                 05BA  ;
                                 05BA
                                 05BA 10$:    ENBINT                           ;ENABLE INTERRUPTS
                0093 C5     94   05BD         CLRB    UCB$B_CEX(R5)            ;FORCE DRIVE FUNCTION
           50   4000 8F     3C   05C1         MOVZWL  #RP_DS_M_ERR,R0          ;SET DRIVE ERROR
                00CE C5     D4   05C6         CLRL    UCB$L_DB_SR(R5)          ;CLEAR SAVED MBA STATUS REGISTER
                     01     88   05CA         BISB    #ERL_M_MEDOFF,-          ; SET FLAG WHICH INDICATES THAT MEDIUM
                00D2 C5          05CC                 UCB$B_DB_ERL(R5)         ;  WAS OFFLINE AT START OF FUNCTION.
                     32     11   05CF         BRB     ERROR                    ;
                                 05D1
                                 05D1
                                 05D1         .ENABL  LSB
                                 05D1
                0117  31  05D1  1229 10$:    BRW     120$                         ; Branch to special condition handler.
                          05D4  1230
                          05D4  1231 ;
                          05D4  1232 ; ENABLE INTERRUPTS
                          05D4  1233 ;
                          05D4  1234
                          05D4  1235 ENBXIT:                                      ;
                          05D4  1236         ENBINT                               ;ENABLE INTERRUPTS
                          05D7  1237
                          05D7  1238 ;
                          05D7  1239 ; RETURN REGISTERS
                          05D7  1240 ;
                          05D7  1241
                          05D7  1242 RETREG:                                      ;RETURN FINAL DEVICE REGISTERS
               34 A3  F7  05D7  1243         CVTLW   RP_ER3(R3),-                 ; Save register after operation.
             00CC C5      05DA  1244                 UCB$W_DB_ER3(R5)
    00C4 C5    38 A3  F7  05DD  1245         CVTLW   RP_ECT(R3),UCB$W_EC1(R5) ;SAVE ECC POSITION REGISTER
    00C6 C5    3C A3  F7  05E3  1246         CVTLW   RP_EC2(R3),UCB$W_EC2(R5) ;SAVE ECC PATTERN REGISTER
                          05E9  1247 ; Here we save the more conservative of the two byte counts contained in
                          05E9  1248 ;       the MBA$L_BCR register.  The high word of this register is the
                          05E9  1249 ;       (negative of the) number of bytes transferred to or from the
                          05E9  1250 ;       drive, while the low word is the (negative of the) number of
                          05E9  1251 ;       bytes transferred to or memory.  On a read, the more conservative
                          05E9  1252 ;       value is that of the number of bytes transferred to memory (low word)
                          05E9  1253 ;       while on a write the more conservative value is the number of
                          05E9  1254 ;       bytes transferred to the drive (high word).  Here we deposit
                          05E9  1255 ;       the entire register into a longword in the UCB.  If the operation
                          05E9  1256 ;       was a read we leave the value as is.  However if the operation
                          05E9  1257 ;       was a write (or anything but a read) we move the high word to
                          05E9  1258 ;       the low word in memory.  All other pieces of this driver use the
                          05E9  1259 ;       low word of this longword as the valid byte count.
                          05E9  1260
               10 A4  D0  05E9  1261         MOVL    MBA$L_BCR(R4),-              ; Save entire byte count register
             00C0 C5      05EC  1262                 UCB$L_DB_BCR(R5)             ;   in the UCB.
         50    58 A5  D0  05EF  1263         MOVL    UCB$L_IRP(R5),R0             ; Retrieve IRP pointer.
                  01  E0  05F3  1264         BBS     #IRP$V_FUNC,-                ; If we had a read operation then
            07 2A A0      05F5  1265                 IRP$W_STS(R0),20$            ;   just branch around since all OK.
             00C2 C5  B0  05F8  1266         MOVW    UCB$L_DB_BCR+2(R5),-         ; If NOT read, then copy high word to
             00C0 C5      05FC  1267                 UCB$W_BCR(R5)                ;   low order word for later use.
                          05FF  1268 20$:
         50    04 A3  D0  05FF  1269         MOVL    RP_DS(R3),R0                 ;GET CONTENTS OF DRIVE STATUS REGISTER
                          0603  1270 ERROR:                                       ;
       51    00CE C5  D0  0603  1271         MOVL    UCB$L_DB_SR(R5),R1           ;RETRIEVE FINAL CONTROLLER STATUS
         52    08 A3  D0  0608  1272         MOVL    RP_ERT(R3),R2                ;GET CONTENTS OF DRIVE ERROR REGISTER 1
 64 A5    0060 8F  B3  060C  1273         BITW    #UCB$M_POWER!-               ;POWER FAIL OR DEVICE TIMEOUT?
                          0612  1274                 UCB$M_TIMOUT,UCB$W_STS(R5) ;
                  BD  12  0612  1275         BNEQ    10$                          ;IF NEQ YES - SPECIAL CONDITION
      0093 C5     0A  91  0614  1276         CMPB    #CDF_WRITECHECK,UCB$B_CEX(R5) ;DRIVE RELATED FUNCTION?
                  2E  1A  0619  1277         BGTRU   30$                          ;IF GTRU YES
      0093 C5     10  91  061B  1278         CMPB    #CDF_READPRESET,UCB$B_CEX(R5) ;OTHER DRIVE RELATED FUNCTION?
                  27  1B  0620  1279         BLEQU   30$                          ;IF EQL YES
                          0622  1280
                          0622  1281 ;
                          0622  1282 ; CONTROLLER RELATED FUNCTION
                          0622  1283 ;
                          0622  1284
 51   000E5FFF 8F  D3  0622  1285         BITL    #MBA$M_ERROR,R1              ;ANY CONTROLLER ERRORS?
```

```
                   6F    13  0629  1286            BEQL    80$                        ;IF EQL NO
          00000000'GF    16  062B  1287            JSB     G^ERL$DEVICERR             ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
        6C 009A C5    0F  E0  0631  1288            BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),90$ ;IF SET, RETRY INHIBITED
     51    0008000B 8F    D3  0637  1289            BITL    #MBA$M_SR_ERCONF!-         ;ERROR CONFIRMATION OR,
                             063E  1290                    MBA$M_SR_ISTO!-            ;INTERFACE SEQUENCE TIMEOUT OR,
                             063E  1291                    MBA$M_SR_PGE!-             ;PROGRAMMING ERROR OR,
                             063E  1292                    MBA$M_SR_RDTO,R1           ;READ TIMEOUT?
                   63    12  063E  1293            BNEQ    90$                        ;IF NEQ YES - FATAL CONTROLLER ERROR
     51    00064FF4 8F    D3  0640  1294            BITL    #MBA$M_SR_DLT!-            ;DATA LATE OR,
                             0647  1295                    MBA$M_SR_INVMAP!-          ;INVALID MAP REGISTER OR,
                             0647  1296                    MBA$M_SR_MAPPE!-           ;MAP REGISTER PARITY ERROR OR,
                             0647  1297                    MBA$M_SR_MBEXC!-           ;MASSBUS EXCEPTION OR,
                             0647  1298                    MBA$M_SR_MCPE!-            ;MASSBUS CONTROL PARITY ERROR OR,
                             0647  1299                    MBA$M_SR_SPE!-             ;SILO PARITY ERROR OR,
                             0647  1300                    MBA$M_SR_MDPE!-            ;MASSBUS DATA PARITY ERROR OR,
                             0647  1301                    MBA$M_SR_MXF!-             ;MISSED TRANSFER OR,
                             0647  1302                    MBA$M_SR_NED!-             ;NONEXISTENT DRIVE OR,
                             0647  1303                    MBA$M_SR_RDS!-             ;READ DATA SUBSTITUTE OR,
                             0647  1304                    MBA$M_SR_WCKLWR!-          ;WRITE CHECK LOWER BYTE OR,
                             0647  1305                    MBA$M_SR_WCKUPR,R1         ;WRITE CHECK UPPER BYTE?
                   31    12  0647  1306            BNEQ    60$                        ;IF NEQ YES - RETRIABLE CONTROLLER ERROR
                             0649  1307
                             0649  1308 ;
                             0649  1309 ; DRIVE RELATED FUNCTION
                             0649  1310 ;
                             0649  1311
                   08    91  0649  1312 30$:      CMPB    #CDF_PACKACK,-             ; Packack function?
              0093 C5        064B  1313                    UCB$B_CEX(R5)              ;
                   0F    12  064E  1314            BNEQ    40$                        ; Branch if not.
          0B 50    0C    E0  0650  1315            BBS     #RP_DS_V_MOL,R0,40$        ; Success if medium online.
                   0F    E1  0654  1316            BBC     #IO$V_INHRETRY,-           ; Branch if retrys not inhibited.
           28 009A C5        0656  1317                    UCB$W_FUNC(R5),65$         ;
                   0B    E5  065A  1318            BBCC    #UCB$V_VALID,-             ; Otherwise, clear software volume
             44 64 A5        065C  1319                    UCB$W_STS(R5),90$          ;  valid and take fatal error path.
          37 50    0E    E1  065F  1320 40$:       BBC     #RP_DS_V_ERR,R0,80$        ;IF CLR, NO DRIVE ERRORS
     00C0 C5    7E A5    AE  0663  1321 50$:       MNEGW   UCB$W_BCNT(R5),UCB$W_BCR(R5) ;RESET BYTE COUNT - NO TRANSFER
           35 00D2 C5    E8  0669  1322            BLBS    UCB$B_DB_ERL(R5),90$       ; Do NOT log error if medium was offline
                             066E  1323                                               ;  at start of function.
          00000000'GF    16  066E  1324            JSB     G^ERL$DEVICERR             ;ALLOCATE AND FILL ERROR MESSAGE BUFFER
        29 009A C5    0F  E0  0674  1325            BBS     #IO$V_INHRETRY,UCB$W_FUNC(R5),90$ ;IF SET, RETRY INHIBITED
          25 50    0C    E1  067A  1326 60$:       BBC     #RP_DS_V_MOL,R0,90$        ;IF CLR, MEDIUM OFFLINE
          21 50    06    E1  067E  1327            BBC     #RP_DS_V_VV,R0,90$         ;IF CLR, INVALID VOLUME
     52    0E07 8F    B3  0682  1328 65$:       BITW    #RP_ER1_M_AOE!-            ;ADDRESS OVERFLOW OR,
                             0687  1329                    RP_ER1_M_IAE!-             ;INVALID ADDRESS OR,
                             0687  1330                    RP_ER1_M_ILF!-             ;ILLEGAL FUNCTION OR,
                             0687  1331                    RP_ER1_M_ILR!-             ;ILLEGAL REGISTER OR,
                             0687  1332                    RP_ER1_M_RMR!-             ;REGISTER MODIFY REFUSE OR,
                             0687  1333                    RP_ER1_M_WLE,R2            ;WRITE LOCK ERROR?
                   1A    12  0687  1334            BNEQ    90$                        ;IF NEQ YES - FATAL DRIVE ERROR
     52    4000 8F    B3  0689  1335            BITW    #RP_ER1_M_UNS,R2           ; Is the drive unsafe?
                   16    12  068E  1336            BNEQ    100$                       ; Branch if so.
                             0690  1337
                             0690  1338 ;
                             0690  1339 ; RETRIABLE ERROR EXIT
                             0690  1340 ;
                             0690  1341
     7E    009C D5    32  0690  1342 70$:       CVTWL   @UCB$L_DPC(R5),-(SP)       ;GET BRANCH DISPLACEMENT
```

```
      009C C5   8E   C0  0695  	ADDL	(SP)+,UCB$L_DPC(R5)	;CALCULATE RETURN ADDRESS - 2
      009C C5   02   C0  069A  80$:	ADDL	#2,UCB$L_DPC(R5)	;SKIP PAST BRANCH DISPLACEMENT WORD
         009C D5    17  069F  	JMP	@UCB$L_DPC(R5)	;RETURN TO DRIVER
                         06A3
                         06A3  ;
                         06A3  ; FATAL CONTROLLER OR DRIVE ERROR
                         06A3  ;
                         06A3
            FD45    31  06A3  90$:	BRW	FATALERR		;
                         06A6
                         06A6  ;
                         06A6  ; Check for unsafe condition and attempt to clear it.
                         06A6  ;
                         06A6
                         06A6  100$:	DSBINT				; Disable interupts.
              05    E1  06AC  	BBC	#UCB$V_POWER,-		; Branch if no power failure occured.
        03 64 A5         06AE  		UCB$W_STS(R5),110$	;
            FF20    31  06B1  	BRW	ENBXIT			; Otherwise, enable interupts and
                         06B4  					;  go process error.
        63    09    9A  06B4  110$:	MOVZBL	#F_DRVCLR!1,RP_CS1(R3)	; Attempt to clear unsafe condition.
                         06B7  	TIMEWAIT -			; Wait for ten microseconds or until
                         06B7  		TIME = #1,-		;  unsafe condition clears.
                         06B7  		BITVAL = #RP_ER1_M_UNS,-;
                         06B7  		SOURCE = RP_ER1(R3),-	;
                         06B7  		CONTEXT = L,-		;
                         06B7  		SENSE = .FALSE.		;
                         06DF  	ENBINT				; Enable interupts.
        52   08 A3    D0  06E2  	MOVL	RP_ER1(R3),R2		; Retrieve error status.
             A7 50    E8  06E6  	BLBS	R0,70$			; Branch if drive is no longer unsafe.
                B8    11  06E9  	BRB	90$			; Otherwise, fatal error.
                         06EB
                         06EB  ;
                         06EB  ; SPECIAL CONDITION (POWER FAILURE OR DEVICE TIME OUT)
                         06EB  ;
                         06EB
  61 64 A5    05    E4  06EB  120$:	BBSC	#UCB$V_POWER,UCB$W_STS(R5),150$ ;IF SET, POWER FAILURE
                         06F0
                         06F0  ;
                         06F0  ; DEVICE TIME OUT
                         06F0  ;
                         06F0
   00000000'GF      16  06F0  	JSB	G^ERL$DEVICTMO		;LOG DEVICE TIME OUT
    53    24 A5     D0  06F6  	MOVL	UCB$L_CRB(R5),R3	;GET ADDRESS OF CRB
    53    2C A3     D0  06FA  	MOVL	CRB$L_INTD+VEC$L_IDB(R3),R3 ;GET ADDRESS OF IDB
    04 A3    55     D1  06FE  	CMPL	R5,IDB$L_OWNER(R3)	;DEVICE OWN CONTROLLER?
             22     12  0702  	BNEQ	140$			;IF NEQ NO
                         0704  	DSBINT				;DISABLE INTERRUPTS
             06     D0  070A  	MOVL	#MBA$M_CR_ABORT!MBA$M_CR_IE,- ;ABORT THE DATA TRANSFER
          04 A4          070C  		MBA$L_CR(R4)		;
                         070E  	WFIKPCH 130$,#15			;WAIT FOR ABORT AND KEEP CHANNEL
                         0718  	IOFORK				;CREATE FORK PROCESS
                         071E  130$:
    04 A4    01     D0  071E  	MOVL	#MBA$M_CR_INIT,MBA$L_CR(R4) ;INITIALIZE ENTIRE MBA
    04 A4    04     D0  0722  	MOVL	#MBA$M_CR_IE,MBA$L_CR(R4) ;ENABLE DEVICE INTERRUPTS
                         0726  140$:	SETIPL	UCB$B_FIPC(R5)		;LOWER TO FORK LEVEL
  50    022C 8F     3C  072A  	MOVZWL	#SS$_TIMEOUT,R0		;SET DEVICE TIMEOUT STATUS
        0080 C5     97  072F  	DECB	UCB$B_ERTCNT(R5)	;ANY ERROR RETRIES REMAINING?
```

```
                    0F   13  0733  1400          BEQL    RESETXFR                  ;IF EQL NO
                             0735  1401          RELCHAN                           ;RELEASE CHANNEL IF OWNED
        64 A5  0040 8F   AA  073B  1402          BICW    #UCB$M_TIMOUT,UCB$W_STS(R5) ;CLEAR TIME OUT STATUS
                  FA68   31  0741  1403          BRW     FDISPATCH                 ;
                             0744  1404
                             0744  1405 ;
                             0744  1406 ; RESET TRANSFER BYTE COUNT TO ZERO
                             0744  1407 ;
                             0744  1408
                             0744  1409 RESETXFR:                                  ;
             53   58 A5  D0  0744  1410          MOVL    UCB$L_IRP(R5),R3          ;RETRIEVE ADDRESS OF I/O PACKET
     00C0 C5   32 A3  AE  0748  1411          MNEGW   IRP$W_BCNT(R3),UCB$W_BCR(R5) ;RESET TRANSFER BYTE COUNT
                  FD2E   31  074E  1412          BRW     FUNCXT                    ;
                             0751  1413
                             0751  1414 ;
                             0751  1415 ; POWER FAILURE
                             0751  1416 ;
                             0751  1417
                             0751  1418 150$:    RELCHAN                           ;RELEASE CHANNEL
             53   58 A5  D0  0757  1419          MOVL    UCB$L_IRP(R5),R3          ;RETRIEVE ADDRESS OF I/O PACKET
          78 A5   2C A3  7D  075B  1420          MOVQ    IRP$L_SVAPTE(R3),UCB$L_SVAPTE(R5) ;RESTORE TRANSFER PARAMETERS
                  F9D4   31  0760  1421          BRW     DB_STARTIO                ;
                             0763  1422          .DSABL  LSB
```

```
                         0763          .SBTTL  RP04/RP05/RP06 CLASSIFY DRIVE TYPE AND SET PARAMETERS
                         0763     ;
                         0763     ; DB_DTYPE - RP04/RP05/RP06 CLASSIFY DRIVE TYPE AND SET PARAMETERS
                         0763     ;
                         0763     ; THIS ROUTINE IS CALLED WHEN AN UNSOLICITED INTERRUPT OCCURS ON A DRIVE, DURING
                         0763     ; SYSTEM INITIALIZATION, AND AT POWER RECOVERY TO DETERMINE THE DRIVE TYPE AND
                         0763     ; SET UNIT PARAMETERS.
                         0763     ;
                         0763     ; INPUTS:
                         0763     ;
                         0763     ;       R3 = ADDRESS OF DRIVE CONTROL REGISTER.
                         0763     ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
                         0763     ;       R5 = DEVICE UNIT UCB ADDRESS.
                         0763     ;
                         0763     ; OUTPUTS:
                         0763     ;
                         0763     ;       THE DRIVE STATUS REGISTER IS INTERROGATED AND UNIT PARAMETERS ARE SET.
                         0763     ;
                         0763
                         0763     DB_DTYPE:                                        ;CLASSIFY DRIVE TYPE AND SET PARAMETERS
            18 A3    DD  0763             PUSHL   RP_DT(R3)                       ;READ DRIVE TYPE REGISTER
6E     FE00 8F       AA  0766             BICW    #^C<^X1FF>,(SP)                 ;CLEAR EXTRANEOUS BITS
52     FBCD CF       9E  076B             MOVAB   DB_DTDESC,R2                    ;GET ADDRESS OF DESCRIPTOR TABLE
       82   6E       B1  0770     10$:    CMPW    (SP),(R2)+                      ;DRIVE TYPE MATCH?
            0E       13  0773             BEQL    20$                             ;IF EQL YES
       52   0D       C0  0775             ADDL    #DB_DTDESCLEN-2,R2              ;ADVANCE TO NEXT ENTRY
            62       B5  0778             TSTW    (R2)                            ;END OF TABLE?
            F4       12  077A             BNEQ    10$                             ;IF NEQ NO
   64 A5    10       AA  077C             BICW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT OFFLINE
       52   0D       C2  0780             SUBL    #DB_DTDESCLEN-2,R2              ;BACK UP TO LAST DRIVE DESCRIPTOR
   41 A5    82       90  0783     20$:    MOVB    (R2)+,UCB$B_DEVTYPE(R5)         ;SET DEVICE TYPE
   44 A5    82       D0  0787             MOVL    (R2)+,UCB$L_DEVDEPEND(R5)  ;SET DISK PACK GEOMETRY
0080 C5     82       D0  078B             MOVL    (R2)+,UCB$L_MAXBLOCK(R5)   ;SET MAXIMUM BLOCKS PER PACK
008C C5     62       D0  0790             MOVL    (R2),UCB$L_MEDIA_ID(R5)    ;SET MEDIA IDENTIFICATION
            8E       D5  0795             TSTL    (SP)+                           ;REMOVE DRIVE TYPE FROM STACK
                     05  0797             RSB                                     ;
```

```
                                   0798              .SBTTL  RP04/05/06 REGISTER DUMP ROUTINE
                                   0798       ;
                                   0798       ; DB_REGDUMP - RP04/05/06 REGISTER DUMP ROUTINE
                                   0798       ;
                                   0798       ; THIS ROUTINE IS CALLED TO SAVE THE CONTROLLER AND DRIVE REGISTERS IN A
                                   0798       ; SPECIFIED BUFFER. IT IS CALLED FROM THE DEVICE ERROR LOGGING ROUTINE AND
                                   0798       ; FROM THE DIAGNOSTIC BUFFER FILL ROUTINE.
                                   0798       ;
                                   0798       ; INPUTS:
                                   0798       ;
                                   0798       ;       R0 = ADDRESS OF REGISTER SAVE BUFFER.
                                   0798       ;       R4 = ADDRESS OF ADAPTER CONFIGURATION REGISTER.
                                   0798       ;       R5 = DEVICE UNIT UCB ADDRESS.
                                   0798       ;
                                   0798       ; OUTPUTS:
                                   0798       ;
                                   0798       ;       THE CONTROLLER AND DRIVE REGISTERS ARE SAVED IN THE SPECIFIED BUFFER.
                                   0798       ;
                                   0798
                                   0798 DB_REGDUMP:                                      ;RP04/05/06 REGISTER DUMP ROUTINE
               80     17     D0    0798         MOVL    #<RP_EC2+4+MBA$L_BCR+4+8>/4,(R0)+ ;INSERT NUMBER OF DEVICE REGISTERS
               80     64     D0    079B         MOVL    MBA$L_CSR(R4),(R0)+             ;SAVE CONFIGURATION REGISTER
            80     04 A4     D0    079E         MOVL    MBA$L_CR(R4),(R0)+              ;SAVE CONTROL REGISTER
         80     00CE C5      D0    07A2         MOVL    UCB$L_DB_SR(R5),(R0)+           ;SAVE STATUS REGISTER
            80     0C A4     D0    07A7         MOVL    MBA$L_VAR(R4),(R0)+             ;SAVE VIRTUAL ADDRESS REGISTER
            80     10 A4     D0    07AB         MOVL    MBA$L_BCR(R4),(R0)+             ;SAVE BYTE COUNT REGISTER
 51   F8 A0    08     09     EF    07AF         EXTZV   #9,#8,-8(R0),R1                 ;GET FINAL MAP REGISTER NUMBER
       80    0800 C441       D0    07B5         MOVL    MBA$L_MAP(R4)[R1],(R0)+         ;SAVE FINAL MAP REGISTER CONTENTS
                      80     D4    07BB         CLRL    (R0)+                           ;ASSUME NO PREVIOUS MAP REGISTER
                      51     D7    07BD         DECL    R1                              ;CALCULATE PREVIOUS MAP REGISTER NUMBER
                      07     19    07BF         BLSS    10$                             ;IF LSS NONE
    FC A0    0800 C441       D0    07C1         MOVL    MBA$L_MAP(R4)[R1],-4(R0)        ;SAVE PREVIOUS MAP REGISTER CONTENTS
               51     10     9A    07C8 10$:    MOVZBL  #<RP_EC2+4>/4,R1                ;SET NUMBER OF DRIVE REGISTERS TO SAVE
          52   0091 C5       9A    07CB         MOVZBL  UCB$B_SLAVE+1(R5),R2            ;GET DRIVE OFFSET CONSTANT
       52    0400 C442       DE    07D0         MOVAL   MBA$L_ERB(R4)[R2],R2            ;GET ADDRESS OF DRIVE REGISTERS
               80     82     D0    07D6 20$:    MOVL    (R2)+,(R0)+                     ;SAVE DRIVE REGISTER
                   FA 51     F5    07D9         SOBGTR  R1,20$                          ;ANY MORE TO SAVE?
                             05    07DC         RSB                                     ;
```

```
                                 07DD              .SBTTL  RP04/RP05/RP06 DISK DRIVE INITIALIZATION
                                 07DD       ;
                                 07DD       ; DB_RPOX_INIT - RP04/RP05/RP06 DISK DRIVE INITIALIZATION
                                 07DD       ;
                                 07DD       ; THIS ROUTINE IS CALLED AT SYSTEM INITIALIZATION AND AT POWER RECOVERY TO SET
                                 07DD       ; DRIVE PARAMETERS AND TO WAIT FOR ONLINE DRIVES TO SPIN UP.
                                 07DD       ;
                                 07DD       ; INPUTS:
                                 07DD       ;
                                 07DD       ;       R4 = ADDRESS OF MBA CONFIGURATION STATUS REGISTER.
                                 07DD       ;       R5 = DEVICE UNIT UCB ADDRESS.
                                 07DD       ;
                                 07DD       ; OUTPUTS:
                                 07DD       ;
                                 07DD       ;       UNIT PARAMETERS ARE ESTABLISHED AND THE DRIVE IS SPUN UP IF IT WAS ONLINE.
                                 07DD       ;
                                 07DD
                                 07DD       DB_RPOX_INIT:                                       ;RP04/RP05/RP06 DISK DRIVE INITIALIZATION
          53    54 A5     3C     07DD               MOVZWL  UCB$W_UNIT(R5),R3           ;GET DRIVE UNIT NUMBER
        0090 C5    53     90     07E1               MOVB    R3,UCB$B_SLAVE(R5)          ;SET SLAVE UNIT NUMBER
                53 20     C4     07E6               MULL    #<1@7>/4,R3                 ;CALCULATE DRIVE OFFSET CONSTANT
        0091 C5    53     90     07E9               MOVB    R3,UCB$B_SLAVE+1(R5)        ;SET DRIVE OFFSET CONSTANT
     53   0400 C443       DE     07EE               MOVAL   MBA$L_ERB(R4)[R3],R3        ;GET ADDRESS OF DRIVE CONTROL REGISTER
                   03     BB     07F4               PUSHR   #^M<R0,R1>                  ;SAVE THESE REGISTERS
                                 07F6               TIMEWAIT #100,#RP_DS_M_DPR,-        ;TRY TO SEIZE DRIVE
                                 07F6                        RP_DS(R3),L
             10 50        E9     0822               BLBC    R0,5$                       ;NO PORT SEIZED
          50    18 A3     D0     0825               MOVL    RP_DT(R3),R0                ;GET DRIVE TYPE
        08 50      08     E1     0829               BBC     #RP_DT_V_DRQ,R0,5$          ;IF CLEAR, LEAVE
                   02     88     082D               BISB    #ERL_M_DUALPORT,-           ; SET FLAG WHICH INDICATES THAT DISK
             00D2 C5             082F                       UCB$B_DB_ERL(R5)            ;  HAS DUAL PORT OPTION
              63   09     9A     0832               MOVZBL  #F_DRVCLR!1,RP_CS1(R3)      ;CLEAR DRIVE
                   03     BA     0835       5$:     POPR    #^M<R0,R1>                  ;RESTORE REGISTERS
          7E    64 A5     3C     0837               MOVZWL  UCB$W_STS(R5),-(SP)         ;SAVE CURRENT UNIT STATUS
                08 A4     DD     083B               PUSHL   MBA$L_SR(R4)                ;READ MBA STATUS REGISTER
    64 A5     0810 8F     AA     083E               BICW    #UCB$M_ONLINE!UCB$M_VALID,UCB$W_STS(R5) ;SET UNIT OFFLINE/INVALID
         32 6E     12     E0     0844               BBS     #MBA$V_SR_NED,(SP),40$      ;IF SET, NONEXISTENT DISK
         64 A5     10     A8     0848               BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
                 FF14     30     084C               BSBW    DB_DTYPE                    ;CLASSIFY DRIVE TYPE
      23 64 A5     04     E1     084F               BBC     #UCB$V_ONLINE,UCB$W_STS(R5),30$ ;IF CLR, UNKNOWN DRIVE TYPE
       1E 04 AE    08     E1     0854               BBC     #UCB$V_VALID,4(SP),30$      ;IF CLR, VOLUME SOFTWARE INVALID
              63   09     9A     0859       10$:    MOVZBL  #F_DRVCLR!1,RP_CS1(R3)      ;CLEAR DRIVE
      52    04 A3  13     78     085C               ASHL    #31-RP_DS_V_MOL,RP_DS(R3),R2 ;MEDIUM ONLINE?
                   08     19     0861               BLSS    20$                         ;IF LSS YES
        00000000'GF       16     0863               JSB     G^EXE$PWRTIMCHK             ;CHECK FOR MAXIMUM TIME EXCEEDED
                ED 50     E8     0869               BLBS    R0,10$                      ;IF LBS MORE TIME TO GO
                   09     11     086C               BRB     30$                         ;
              63   13     9A     086E       20$:    MOVZBL  #F_PACKACK!1,RP_CS1(R3)     ;ACKNOWLEDGE PACK
    64 A5     0800 8F     A8     0871               BISW    #UCB$M_VALID,UCB$W_STS(R5) ;SET VOLUME SOFTWARE VALID
              63   08     9A     0877       30$:    MOVZBL  #F_RELEASE!1,RP_CS1(R3)     ;Clear drive and release port
    08 A4     8E   8E     C9     087A       40$:    BISL3   (SP)+,(SP)+,MBA$L_SR(R4)    ;CLEAR MBA STATUS
                          05     087F               RSB                                 ;
                          0880         .SBTTL  RP04/RP05/RP06 UNSOLICITED INTERRUPT ROUTINE
                          0880  ;
                          0880  ; DB_UNSOLNT - RP04/RP05/RP06 UNSOLICITED INTERRUPT ROUTINE
                          0880  ;
                          0880  ; THIS ROUTINE IS CALLED WHEN AN UNSOLICITED ATTENTION CONDITION IS DETECTED FOR
                          0880  ; AN RP04, RP05, OR RP06 DRIVE.
                          0880  ;
                          0880  ; INPUTS:
                          0880  ;
                          0880  ;       R4 = ADDRESS OF CONFIGURATION STATUS REGISTER.
                          0880  ;       R5 = DEVICE UNIT UCB ADDRESS.
                          0880  ;
                          0880  ; OUTPUTS:
                          0880  ;
                          0880  ;       IF VOLUME VALID IS CLEAR, THEN SOFTWARE VOLUME VALID IS CLEARED. THE
                          0880  ;       UNIT STATUS IS CHANGED TO ONLINE AND THE DRIVE TYPE AND PARAMETERS ARE
                          0880  ;       CLASSIFIED.
                          0880  ;
                          0880
                          0880  DB_UNSOLNT:                                       ;RP04/RP05/RP06 UNSOLICITED INTERRUPTS
   53   0091 C5      9A   0880          MOVZBL  UCB$B_SLAVE+1(R5),R3      ;GET DRIVE OFFSET CONSTANT
 53   0400 C443      DE   0885          MOVAL   MBA$L_ERB(R4)[R3],R3      ;GET ADDRESS OF DRIVE CONTROL REGISTER
     64 A5    10     A8   088B          BISW    #UCB$M_ONLINE,UCB$W_STS(R5) ;SET UNIT ONLINE
            FED1     30   088F          BSBW    DB_DTYPE                  ;CLASSIFY DRIVE TYPE
  1F 64 A5    04     E1   0892          BBC     #UCB$V_ONLINE,UCB$W_STS(R5),10$ ;IF CLR, UNKNOWN DRIVE TYPE
  20 64 A5    08     E1   0897          BBC     #UCB$V_VALID,UCB$W_STS(R5),20$ ;IF CLR, VOLUME SOFTWARE INVALID
52   04 A3    13     78   089C          ASHL    #31-RP_DS_V_MOL,RP_DS(R3),R2 ;MEDIUM ONLINE?
              13     18   08A1          BGEQ    10$                       ;IF GEQ NO
  07 64 A5    08     E1   08A3          BBC     #UCB$V_BSY,UCB$W_STS(R5),5$ ;We know the drive is online; thus,
    0093 C5   08     91   08A8          CMPB    #CDF_PACKACK,UCB$B_CEX(R5)  ;if busy doing a PACKACK function,
              0D     13   08AD          BEQL    20$                         ;then don't clear software valid.
52   04 A3    19     78   08AF  5$:     ASHL    #31-RP_DS_V_VV,RP_DS(R3),R2 ;VOLUME VALID?
              06     19   08B4          BLSS    20$                       ;IF LSS YES
64 A5   0800 8F      AA   08B6  10$:    BICW    #UCB$M_VALID,UCB$W_STS(R5) ;CLEAR SOFTWARE VOLUME VALID
                     05   08BC  20$:    RSB                               ;
                          08BD  DB_END:                                   ;ADDRESS OF LAST LOCATION IN DRIVER
                          08BD
                          08BD          .END
```

```
$$$                       = 00000020 R      02    DPT$C_LENGTH                = 00000038
$$OP                      = 00000002              DPT$C_VERSION               = 00000004
ACP$ACCESS                  ********    X   03    DPT$INITAB                    00000038 R      02
ACP$DEACCESS                ********    X   03    DPT$M_SVP                   = 00000002
ACP$MODIFY                  ********    X   03    DPT$REINITAB                  0000006A R      02
ACP$MOUNT                   ********    X   03    DPT$TAB                       00000000 R      02
ACP$READBLK                 ********    X   03    DRVCLR                        00000209 R      03
ACP$WRITEBLK                ********    X   03    DT$_RP04                    = 00000003
APPLY_ECC                   000002FE R      03    DT$_RP05                    = 00000004
AT$_MBA                   = 00000000              DT$_RP06                    = 00000005
CDF_DRVCLR                = 00000004              DYN$C_DDB                   = 00000006
CDF_NOP                   = 00000005              DYN$C_DPT                   = 0000001E
CDF_OFFSET                = 00000006              DYN$C_UCB                   = 00000010
CDF_PACKACK               = 00000008              ECC                           000002AE R      03
CDF_READDATA              = 0000000C              EMB$L_DV_REGSAV             = 0000004E
CDF_READHEAD              = 0000000E              ENBXIT                        000005D4 R      03
CDF_READPRESET            = 00000010              ERL$DEVICERR                  ********    X   03
CDF_RECAL                 = 00000003              ERL$DEVICTMO                  ********    X   03
CDF_RETCENTER             = 00000007              ERL_M_DUALPORT              = 00000002
CDF_SEARCH                = 00000009              ERL_M_ECC_DEFER             = 00000004
CDF_SEARCHA               = 00000011              ERL_M_MEDOFF                = 00000001
CDF_SEEK                  = 00000002              ERL_V_DUALPORT              = 00000001
CDF_UNLOAD                = 00000001              ERL_V_ECC_DEFER             = 00000002
CDF_WRITECHECK            = 0000000A              ERROR                         00000603 R      03
CDF_WRITECHECKH           = 0000000F              EXE$GL_TENUSEC                ********    X   03
CDF_WRITEDATA             = 0000000B              EXE$GL_UBDELAY                ********    X   03
CDF_WRITEHEAD             = 0000000D              EXE$IOFORK                    ********    X   03
CHECKRETRY                  0000026D R      03    EXE$LCLDSKVALID               ********    X   03
CHECKTAB                    00000038 R      03    EXE$ONEPARM                   ********    X   03
CHECKXT                     00000286 R      03    EXE$PWRTIMCHK                 ********    X   03
CRB$L_INTD                = 00000024              EXE$SENSEMODE                 ********    X   03
DATACHECK                   0000023A R      03    EXE$SETCHAR                   ********    X   03
DB$DDT                      00000000 RG     03    EXE$ZEROPARM                  ********    X   03
DB_DTDESC                   0000003C R      03    EXFNC                         00000581 R      03
DB_DTUESCLEN              = 0000000F              FATALERR                      000003EB R      03
DB_DTYPE                    00000763 R      03    FDISPATCH                     000001AC R      03
DB_END                      0000088D R      03    FEX                           000004C3 R      03
DB_FUNCTABLE                000000A3 R      03    FTAB                          00000089 R      03
DB_REGDUMP                  00000798 R      03    FUNCTAB_LEN                 = 00000094
DB_RPOX_INIT                000007DD R      03    FUNCXT                        0000047F R      03
DB_STARTIO                  00000137 R      03    F_DRVCLR                    = 00000008
DB_UNSOLNT                  00000880 R      03    F_NOP                       = 00000000
DC$_DISK                  = 00000001              F_OFFSET                    = 0000000C
DDB$K_PACK                = 00000001              F_PACKACK                   = 00000012
DDB$L_ACPD                = 00000010              F_READDATA                  = 00000038
DDB$L_DDT                 = 0000000C              F_READHEAD                  = 0000003A
DEFER_ECC                   00000327 R      03    F_READPRESET                = 00000010
DEV$M_AVL                 = 00040000              F_RECAL                     = 00000006
DEV$M_DIR                 = 00000008              F_RELEASE                   = 0000000A
DEV$M_DUA                 = 00008000              F_RETCENTER                 = 0000000E
DEV$M_ELG                 = 00400000              F_SEARCH                    = 00000018
DEV$M_FOD                 = 00004000              F_SEARCHA                   = 00000018
DEV$M_IDV                 = 04000000              F_SEEK                      = 00000004
DEV$M_NNM                 = 00000200              F_UNLOAD                    = 00000002
DEV$M_ODV                 = 08000000              F_WRITECHECK                = 00000028
DEV$M_RND                 = 10000000              F_WRITECHECKH               = 0000002A
DEV$M_SHR                 = 00010000              F_WRITEDATA                 = 00000030
```

```
F_WRITEHEAD                 = 00000032                 IRP$V_PHYSIO                = 00000008
GO                            000004E2 R       03      IRP$W_BCNT                  = 00000032
IDB$L_OWNER                 = 00000004                 IRP$W_FUNC                  = 00000020
IMMED                         00000508 R       03      IRP$W_STS                   = 0000002A
IO$M_DATACHECK              = 00004000                 LDCYL                         0000057B R       03
IO$V_DATACHECK              = 0000000E                 MASKH                       = 00000008
IO$V_INHRETRY               = 0000000F                 MASKL                       = 04000000
IO$V_INHSEEK                = 0000000C                 MBA$L_BCR                   = 00000010
IO$_ACCESS                  = 00000032                 MBA$L_CR                    = 00000004
IO$_ACPCONTROL              = 00000038                 MBA$L_CSR                   = 00000000
IO$_AVAILABLE               = 00000011                 MBA$L_ERB                   = 00000400
IO$_CREATE                  = 00000033                 MBA$L_MAP                   = 00000800
IO$_DEACCESS                = 00000034                 MBA$L_SR                    = 00000008
IO$_DELETE                  = 00000035                 MBA$L_VAR                   = 0000000C
IO$_DRVCLR                  = 00000004                 MBA$M_CR_ABORT              = 00000002
IO$_MODIFY                  = 00000036                 MBA$M_CR_IE                 = 00000004
IO$_MOUNT                   = 00000039                 MBA$M_CR_INIT               = 00000001
IO$_NOP                     = 00000000                 MBA$M_ERROR                 = 000E5FFF
IO$_OFFSET                  = 00000006                 MBA$M_SR_DLT                = 00000800
IO$_PACKACK                 = 00000008                 MBA$M_SR_ERCONF             = 00000008
IO$_READHEAD                = 0000000E                 MBA$M_SR_INVMAP             = 00000010
IO$_READLBLK                = 00000021                 MBA$M_SR_ISTO               = 00000002
IO$_READPBLK                = 0000000C                 MBA$M_SR_MAPPE              = 00000020
IO$_READPRESET              = 00000019                 MBA$M_SR_MBEXC              = 00000080
IO$_READVBLK                = 00000031                 MBA$M_SR_MCPE               = 00020000
IO$_RECAL                   = 00000003                 MBA$M_SR_MDPE               = 00000040
IO$_RELEASE                 = 00000005                 MBA$M_SR_MXF                = 00000100
IO$_RETCENTER               = 00000007                 MBA$M_SR_NED                = 00040000
IO$_SEARCH                  = 00000009                 MBA$M_SR_PGE                = 00080000
IO$_SEEK                    = 00000002                 MBA$M_SR_RDS                = 00000004
IO$_SENSECHAR               = 0000001B                 MBA$M_SR_RDTO               = 00000001
IO$_SENSEMODE               = 00000027                 MBA$M_SR_SPE                = 00004000
IO$_SETCHAR                 = 0000001A                 MBA$M_SR_WCKLWR             = 00000200
IO$_SETMODE                 = 00000023                 MBA$M_SR_WCKUPR             = 00000400
IO$_UNLOAD                  = 00000001                 MBA$V_SR_NED                = 00000012
IO$_VIRTUAL                 = 0000003F                 NOP                           00000209 R       03
IO$_WRITECHECK              = 0000000A                 NORMAL                        00000283 R       03
IO$_WRITECHECKH             = 00000018                 OFF                           0000032C R       03
IO$_WRITEHEAD               = 0000000D                 OFFSET                        00000209 R       03
IO$_WRITELBLK               = 00000020                 OFFSETERR                     000003B3 R       03
IO$_WRITEPBLK               = 0000000B                 OFFSIZ                      = 00000008
IO$_WRITEVBLK               = 00000030                 OFFTAB                        0000009B R       03
IOC$APPLYECC                  ********    X    03      PACKACK                       00000203 R       03
IOC$DIAGBUFILL                ********    X    03      POSIT                         00000575 R       03
IOC$LOADMBAMAP                ********    X    03      PR$_IPL                     = 00000012
IOC$MNTVER                    ********    X    03      READDATA                      0000021C R       03
IOC$RELCHAN                   ********    X    03      READHEAD                      0000021C R       03
IOC$REQCOM                    ********    X    03      READPRESET                    00000209 R       03
IOC$REQPCHANL                 ********    X    03      RECAL                         00000209 R       03
IOC$RETURN                    ********    X    03      RELEASE                       00000209 R       03
IOC$UPDATRANSP                ********    X    03      RESETXFR                      00000744 R       03
IOC$WFIKPCH                   ********    X    03      RETCENTER                     00000209 R       03
IRP$L_MEDIA                 = 00000038                 RETREG                        000005D7 R       03
IRP$L_SVAPTE                = 0000002C                 RETRY                         000002AB R       03
IRP$S_FCODE                 = 00000006                 RETRYERR                      000003BE R       03
IRP$V_FCODE                 = 00000000                 RP_AS                         00000010
IRP$V_FUNC                  = 00000001                 RP_CC                         0000002C
```

```
RP_CS1                          00000000                 SS$_DRVERR                   = 0000008C
RP_CS1_M_GO                   = 00000001                 SS$_FORMAT                   = 000000BC
RP_DA                           00000014                 SS$_IVADDR                   = 00000134
RP_DC                           00000028                 SS$_MEDOFL                   = 000001A4
RP_DS                           00000004                 SS$_NONEXDRV                 = 000001C4
RP_DS_M_DPR                   = 00000100                 SS$_NORMAL                   = 00000001
RP_DS_M_ERR                   = 00004000                 SS$_OPINCOMPL                = 000002D4
RP_DS_V_ERR                   = 0000000E                 SS$_PARITY                   = 000001F4
RP_DS_V_MOL                   = 0000000C                 SS$_TIMEOUT                  = 0000022C
RP_DS_V_VV                    = 00000006                 SS$_UNSAFE                   = 0000023C
RP_DT                           00000018                 SS$_VOLINV                   = 00000254
RP_DT_V_DRQ                   = 0000000B                 SS$_WASECC                   = 00000639
RP_ECT                          00000038                 SS$_WRITLCK                  = 0000025C
RP_EC2                          0000003C                 TRANNOCH                       00000230 R      03
RP_ER1                          00000008                 TRANRQCH                       0000022A R      03
RP_ER1_M_AOE                  = 00000200                 TRANXT                         00000289 R      03
RP_ER1_M_DCK                  = 00008000                 UCB$B_CEX                    = 00000093
RP_ER1_M_DTE                  = 00001000                 UCB$B_DB_ERL                   000000D2
RP_ER1_M_ECH                  = 00000040                 UCB$B_DEVCLASS               = 00000040
RP_ER1_M_FER                  = 00000010                 UCB$B_DEVTYPE                = 00000041
RP_ER1_M_HCE                  = 00000080                 UCB$B_DIPL                   = 0000005E
RP_ER1_M_HCRC                 = 00000100                 UCB$B_ERTCNT                 = 00000080
RP_ER1_M_IAE                  = 00000400                 UCB$B_ERTMAX                 = 00000081
RP_ER1_M_ILF                  = 00000001                 UCB$B_FEX                    = 00000092
RP_ER1_M_ILR                  = 00000002                 UCB$B_FIPL                   = 0000000B
RP_ER1_M_OPI                  = 00002000                 UCB$B_OFFNDX                 = 000000CA
RP_ER1_M_PAR                  = 00000008                 UCB$B_OFFRTC                 = 000000CB
RP_ER1_M_RMR                  = 00000004                 UCB$B_SECTORS                = 00000044
RP_ER1_M_UNS                  = 00004000                 UCB$B_SLAVE                  = 00000090
RP_ER1_M_WCF                  = 00000020                 UCB$K_DB_LENGTH              = 000000D6
RP_ER1_M_WLE                  = 00000800                 UCB$K_LCC_DISK_LENGTH        = 000000CC
RP_ER1_V_FER                  = 00000004                 UCB$L_BCR                    = 000000C0
RP_ER1_V_HCE                  = 00000007                 UCB$L_CRB                    = 00000024
RP_ER1_V_HCRC                 = 00000008                 UCB$L_DB_BCR                 = 000000C0
RP_ER1_V_OPI                  = 0000000D                 UCB$L_DB_SR                    000000CE
RP_ER1_V_UNS                  = 0000000E                 UCB$L_DEVCHAR                = 00000038
RP_ER1_V_WLE                  = 0000000B                 UCB$L_DEVCHAR2               = 0000003C
RP_ER2                          00000020                 UCB$L_DEVDEPEND              = 00000044
RP_ER3                          00000034                 UCB$L_DPC                    = 0000009C
RP_ER3_V_SKI                  = 0000000E                 UCB$L_IRP                    = 00000058
RP_LA                           0000001C                 UCB$L_MAXBLOCK               = 000000B0
RP_MR                           0000000C                 UCB$L_MEDIA_ID               = 0000008C
RP_OF                           00000024                 UCB$L_SVAPTE                 = 00000078
RP_OF_M_DCK                   = 00000100                 UCB$M_ONLINE                 = 00000010
RP_OF_M_ECI                   = 00000800                 UCB$M_POWER                  = 00000020
RP_OF_M_FMT                   = 00001000                 UCB$M_TIMOUT                 = 00000040
RP_OF_M_HCI                   = 00000400                 UCB$M_VALID                  = 00000800
RP_OF_V_DCK                   = 00000008                 UCB$V_BSY                    = 00000008
RP_OF_V_ECI                   = 0000000B                 UCB$V_ECC                    = 00000000
RP_SN                           00000030                 UCB$V_ONLINE                 = 00000004
SEARCH                          00000209 R      03       UCB$V_POWER                  = 00000005
SEARCHA                         00000552 R      03       UCB$V_VALID                  = 0000000B
SEEK                            00000209 R      03       UCB$W_BCNT                   = 0000007E
SEIZE                           00000522 R      03       UCB$W_BCR                    = 000000C0
SIZ...                        = 00000001                 UCB$W_DA                     = 000000BC
SS$_CTRLERR                   = 00000054                 UCB$W_DB_ER3                   000000CC
SS$_DATACHECK                 = 0000005C                 UCB$W_DC                     = 000000BE
```

```
UCB$W_DEVBUFSIZ                 = 00000042
UCB$W_DEVSTS                    = 00000068
UCB$W_EC1                       = 000000C4
UCB$W_EC2                       = 000000C6
UCB$W_FUNC                      = 0000009A
UCB$W_OFFSET                    = 000000C8
UCB$W_STS                       = 00000064
UCB$W_UNIT                      = 00000054
UNLOAD                            000001FB R      03
VEC$L_IDB                       = 00000008
WRITECHECK                        00000210 R      03
WRITECHECKH                       00000210 R      03
WRITEDATA                         00000217 R      03
WRITEHEAD                         00000217 R      03
XFER                              00000566 R      03

                                               +----------------+
                                               ! Psect synopsis !
                                               +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (     0.)  00 (  0.)  NOPIC    USR    CON    ABS    LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                           000000D6  (   214.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$$105_PROLOGUE                 00000070  (   112.)  02 (  2.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$$$115_DRIVER                   000008BD  (  2237.)  03 (  3.)  NOPIC    USR    CON    REL    LCL NOSHR   EXE   RD     WRT NOVEC LONG

                                            +------------------------+
                                            ! Performance indicators !
                                            +------------------------+

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        32    00:00:00.06    00:00:00.29
Command processing                   129    00:00:00.40    00:00:01.70
Pass 1                               585    00:00:19.40    00:01:11.47
Symbol table sort                      0    00:00:02.51    00:00:07.70
Pass 2                               285    00:00:04.18    00:00:17.00
Symbol table output                   43    00:00:00.21    00:00:00.77
Psect synopsis output                  2    00:00:00.01    00:00:00.01
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                1078    00:00:26.77    00:01:38.94

The working set limit was 2100 pages.
151533 bytes (296 pages) of virtual memory were used to buffer the intermediate code.
There were 120 pages of symbol table space allocated to hold 2322 non-local and 65 local symbols.
1590 source lines were read in Pass 1, producing 22 object records in Pass 2.
48 pages of virtual memory were used to define 45 macros.
```

```
                                    +----------------------------+
                                    ! Macro library statistics !
                                    +----------------------------+

Macro library name                  Macros defined
------------------                  --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                  30
_$255$DUA28:[SYSLIB]STARLET.MLB;2               10
TOTALS (all libraries)                          40

2486 GETS were required to define 40 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:DBDRIVER/OBJ=OBJ$:DBDRIVER MSRC$:DBDRIVER/UPDATE=(ENH$:DBDRIVER)+EXECML$/LIB
```

0108 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY